**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроніки
(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації
(повна назва кафедри)

«На правах рукопису»
УДК 004.738.5

«До захисту допущено»
Завідувач кафедри
_____ _____
(підпис) (ініціали, прізвище)
“ 9 ” грудня 2019 р.

# Магістерська дисертація

спеціальність 171 Електроніка
(код і назва спеціальності)

на тему:«Дослідження засобів генерації ключів в системах з криптографічним захистом».

Виконав студент VI курсу, групи ДВ-82мп
(шифр групи)

Горобченко Микита Олегович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник доцент, д.т.н., проф. Савченко Ю.Г.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України**
**«Київський політехнічний інститут**
**імені Ігоря Сікорського»**

Інститут (факультет)_____Факультет електроніки_____
(повна назва)

Кафедра_____Кафедра звукотехніки та реєстрації інформації_____
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма)171 Електроніка (Електронні системи мультимедіа та засоби Інтернету речей)_____
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ _____
(підпис) (ініціали, прізвище)
«20»_____жовтня___2018 р.

**ЗАВДАННЯ**
**на магістерську дисертацію студенту**
_____Горобченко Микити Олеговича_____
(прізвище, ім'я, по батькові)

1. Тема дисертації Дослідження засобів генерації ключів в системах з криптографічним захистом.
науковий керівник дисертації Савченко Юлій Григорович д.т.н.,професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2019р. №3859-с

2. Строк подання студентом дисертації 09.12.2019р. _____

3. Об'єкт дослідження: Об'єктом дослідження є процедури генерації псевдовипадкових бінарних послідовностей

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою)1) алгоритми генерації на основі використання регістрів зсуву із зворотніми зв'язками по модулю 2

5. Перелік завдань, які потрібно розробити: Визначення недоліків регістрових структур та критеріїв якості отриманих послідовностей. Аналіз існуючих тестів на випадковість. Дослідження шляхів

вдосконалення алгоритмів генерації ключів та засобів програмної та апаратної реалізації процедур генерації.

6. Перелік графічного (ілюстративного) матеріалу 15 рисунків у роботі, 19 таблиць, 1 презентація, 10 слайдів.

7. Орієнтовний перелік публікацій 1.) «Особливості генерації ключів шифрування в захищених системах зв'язку з асиметричним шифруванням» // II ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНІ ТЕХНОЛОГІЇ КІНО ТА АУДІОВІЗУАЛЬНИХ СИСТЕМ» 2019р. -С.81-82 2.) «Особливості шифрування даних в захищених системах зв'язку » // II ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНІ ТЕХНОЛОГІЇ КІНО ТА АУДІОВІЗУАЛЬНИХ СИСТЕМ», 2019р., -С.82-83.

8. Консультанти розділів дисертації*

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| | | | |

9. Дата видачі завдання 20.10.2018

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Строк виконання етапів магістерської дисертації | Примітка |
|---|---|---|---|
| | Написання першого розділу: «Проблеми генерації псевдовипадкових бінарних послідовностей (ПВБП) та шляхи їх вирішення». | 11.10.2019 | |
| | Написання другого розділу:«Статичні характеристики ПВБП, оцінка якості послідовностей». | 23.10.2019 | |
| | Написання третього розділу: «Реалізація алгоритмів генерації на основі використання регістрових структур». | 04.11.2019 | |
| | Написання четвертого розділу: «Узагальнений підхід на основі моделі цифрових автоматів». | 12.11.2019 | |
| | Написання п'ятого розділу «розроблення стартап-проекту» та підготовка матеріалів до друку та оформлення пояснювальної записки. | 26.11.2019 | |
| | Підготовка презентації для доповіді | 30.11.2019 | |

Студент _____ М.О. Горобченко
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Ю. Г. Савченко
(підпис) (ініціали, прізвище)

УДК 004.738.5

## SUMMARY

Mykyta Olegovich Gorobchenko’s attestation master's thesis on "Generation of keys in telecommunication systems with asymmetric encryption" contains 112 pages, 20 figures,24 tables, 24 links.

CRYPTOGRAPHY, SHIFT REGISTER, Kiv-FILTER, ENCODING, DIAGNOSTICS, ENCRYPTION, AUTOMATIC MODELS, DECODER MEHITA, CYCLIC CODE, SIGNATUDE ANALYSIS, LINEAR DIGITAL FILTER, FIBONACCI CONFIGURATION.

Object of researching - pseudorandom sequence generators in telecommunication systems.

The research method is a theoretical analysis of different schemes of pseudorandom sequences generation as well as their practical implementation, investigation of the work of shift registers with linear feedback.

The methods of efficient generation of pseudorandom binary sequences on two-shift shift registers are analyzed, problems of extension of generation of such schemes are investigated, polynomials and their selection for use in such registers are investigated. The selection of polynomials and the implementation of such a register with module shift two on these polynomials are also practically considered in the paper.

УДК 004.738.5

# РЕФЕРАТ

Магістерська дисертація: 112 с., 20 рис., 26 табл., 24 посилання.

КРИПТОГРАФІЯ, РЕГІСТР ЗСУВУ, КІВ-ФІЛЬТР, КОДУВАННЯ, ДІАГНОСТИКА, ШИФРУВАННЯ, АВТОМАТНІ МОДЕЛІ ДЕКОДЕР МЕГІТА, ЦИКЛІЧНИЙ КОД, СИГНАТУРНИЙ АНАЛІЗАТОР, ЛІНІЙНИЙ ЦИФРОВИЙ ФІЛЬТР, КОНФІГУРАЦІЯ ФІБОНАЧЧІ.

**Актуальність теми.** З розвитком систем зв'язку витікає проблема захисту інформації від різних ризиків, а також проблема діагностування цифрової апаратури. Це не тільки захист від доступу сторонніх осіб, а також від помилок. В цих сферах знайшли своє місце і широко використовуються генератори псевдовипадкових послідовностей на основі регістрових структур. Такі генератори використовуються в великій кількості в різноманітних захищених телекомунікаційних систем.

Не можна також забувати про діагностування цифрових схем у складі великих комплексів, де лінійні цифрові фільтри дозволяють будувати прості і надзвичайно компактні пристрої для ефективного і швидкого знаходження несправностей.

**Зв'язок роботи з науковими програмами, планами, темами**. Напрямок досліджень пов'язаний з науковою тематикою і темами навчального процесу кафедри звукотехніки та реєстрації інформації факультету електроніки Національного технічного університету України "Київський політехнічний інститут".

**Метою роботи** є всебічний аналіз застосування моделей цифрових автоматів та побудови генераторів ПВБП на їх основі.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- докладне ознайомлення з функціями регістрів зсуву зі зворотними зв’язками та їх можливостями;
- дослідження основних сфер застосування ПВБП;
- дослідження основних засобів діагностування несправностей в цифрових системах та участю лінійних цифрових фільтрів в них;
- дослідження принципів побудови генераторів псевдовипадкових послідовностей на регістрах зсуву зі зворотнім зв’язком по модулю два, методи підбору поліномів для них та перспектив розширення можливостей таких генераторів;
- визначення основних можливостей таких фільтрів для систем шифрування інформації;
- аналіз вимог до ПВБП, генераторів, тестів NIST

**Об’єктом дослідження** є оптимізація застосування регістрів зсуву, зокрема в генераторах послідовностей.

**Предметом дослідження** є структурні схеми побудови лінійних цифрових фільтрів з застосуванням регістрів зсуву зі зворотними зв’язками.

**Методом дослідження**, що використовується для виконання поставлених задач, є порівняння різних принципів побудови систем на основі регістрів зсуву зі зворотними зв’язками на основі літературних джерел з даної тематики та їх аналіз.

**Наукова новизна отриманих результатів**.

1. Запропоновані нові реалізації моделей цифрових автоматів на основі регістрів зі зворотнім зв’язком для генерації псевдовипадкових послідовностей.
2. Запропоновано нові поліноми для регістрів зсуву зі зворотнім зв’язком .
3. Запропонована програмна реалізація регістру зсуву зі зворотнім звязком по модулю два.

.

**Практична значимість.** Отримані в результаті виконання роботи дані можуть бути використані для подальшого вдосконалення систем кодування та шифрування інформації. В створенні більш досконалих систем діагностики цифрових пристроїв.

**Структура та обсяг дисертації**. Магістерська дисертація складається зі вступу, шести розділів, висновків та одного додатку. Вона містить 112 сторінок друкованого тексту, 20 рисунків та 26 таблиць, перелік посилань з 24 джерел та 2 додатки на 14 сторінках.

**ЗМІСТ**

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

| | | |
|---|---|---|
| "HP" | – | "Hewlett – Packard" |
| LFSR | – | Linear Feedback Shift Register (регістр зсуву зі зворотним зв’язком) |
| БР | – | Буферний Регістр |
| БЧХ | – | код Боуза-Чоудхурі-Хоквінчема |
| ВУ | – | Вузол Управління |
| ДМ | – | декодер Мегіта |
| ЕОМ | – | Електронна обчислювальна Машина |
| ЗЗ | – | Зворотний Зв’язок |
| КІВ | – | Кінцевий Імпульсний Відгук |
| ОЗП | – | Оперативний Запам’ятовуючий Пристрій |
| ПЗП | – | Постійний Запам’ятовуючий Пристрій |
| ГП | – | Генераторний Поліном |
| СА | – | Сигнатурний Аналізатор |
| СІ | – | Синхронізуючий Імпульс |
| СП | – | Синдромний Поліном |
| ПВП | – | Псевдо Випадкова Послідовність |
| ГПВП | – | Генератор Псевдо Випадкової Послідовності |

# ВСТУП

В сучасному світі все більшого розповсюдження та значення в усіх сферах життєдіяльності набувають різноманітні системи пов’язані з передачею та зберіганням інформації. Така інформація часто буває конфіденційною або не призначеною для доступу сторонніх осіб, тому гостро постає питання про захист такої інформації шляхом шифрування. Існує досить велика кількість варіантів реалізацій для вирішення таких проблем, але більшість з них пов’язана з проблемою генерації псевдовипадкових бінарних послідовностей, оскільки від якості таких послідовностей залежить те, наскільки легко буде розшифрувати зашифровану інформацію. Саме тому  питання про те яким чином можливо генерувати якісні послідовності, та яким чином оцінювати якість таких послідовностей і вважаю актуальним.

В  в дисертації вивчаються генератори псевдовипадкових бінарних послідовностей зі зворотнім зв’язком по модулю 2 а також поліноми зворотніх звязків. Передбачається підібрати поліном зворотнього зв’язку та програмно реалізувати генератор ПВБП з використанням цього поліному.

Для реалізації перерахованого необхідно:

1. Вивчити вимоги, що висуваються до ПВБП та дослідити методи їх оцінки.
2. Підібрати поліном зворотнього зв’язку для генератора ПВБП за допомогою спеціальних програмних пакетів.
3. Програмно реалізувати генератор ПВБП з використанням цього поліному.

# 1 ПРОБЛЕМА ГЕНЕРАЦІЇ ПСЕВДОВИПАДКОВИХ БІНАРНИХ ПОСЛІДОВНОСТЕЙ (ПВБП) ТА ШЛЯХИ ЇХ ВИРІШЕННЯ

## 1.1 Сфери застосування ПВБП

Генератори псевдовипадкових послідовностей (ПВП) широко застосовуються в техніці. Одними з основних областей їх застосування є:

- Коди для виявлення та виправлення помилок;
- тестування цифрових пристроїв та інтегральних схем;
- шифрування та різні види захисту інформації

Якісні псевдовипадкові послідовності, будучи по суті детермінованими, мають всі властивості реалізацій істинно випадкових процесів та успішно їх заміняють, оскільки випадкові послідовності надзвичайно важко формувати. Від якості використаних генераторів залежить якість отриманих результатів. Цю обставину підкреслює відомий афоризм Роберта Р. Кав'ю: "*Генерація випадкових чисел надто важлива, щоб залишати її на волю випадку*". Джерела справжніх випадкових чисел знайти важко. Фізичні шуми, такі як детектори змініонізуючої радіації, дробовий шум в резисторіабо космічне випромінюванняможуть бути такими джерелами. Однак застосовуються такі пристроїв додатках мережевої безпеки рідко. Складності також визивають грубі атакина подібні пристрої. Альтернативним вирішенням є створеннядеякого набору з великої кількостівипадкових чиселі опублікування його в деякому словнику. Проте, і такі набори забезпечуютьдуже обмежене джерело чиселу порівнянні з тою кількістю, яка необхідна додаткам мережевої безпеки. Хоч набори з цих таблицьдійсно забезпечують статистичну випадковість, вони не достатньо випадкові, оскільки зловмисник може отримати копію словника.

Криптографічні додатки використовують для генерування випадкових чисел особливі алгоритми. Ці алгоритми заздалегідь визначені та генерують послідовність чисел, яка теоретично не може бути статистично випадковою. В той же час, якщо обирати хороший алгоритм, отримана числова послідовність

буде проходити більшість тестів на випадковість. Такі числа і є псевдовипадковими числами.

Жодний детермінований алгоритм не може генерувати повністю випадкові числа, він може тільки апроксимувати деякі властивості випадкових чисел. Як зауважив Джон фон Нейман, "*усякий, хто має слабкість до арифметичних методів отримання випадкових чисел, грішний поза всякими сумнівами*".

Будь-який ГПВП з обмеженими ресурсами рано чи пізно"зациклюється"– починає повторювати одну й ту ж послідовність чисел. Довжина циклів ГПВП залежить від самого генератора та в середньому складає приблизно $2^{n/2}$, де $n$– розмір внутрішнього стану в бітах, хоч лінійні конгруентні та лінійні ГПВП мають максимальний цикл порядку $2^n$. Якщо утворена ГПВП послідовність зходиться до дуже коротких циклів, то такий ГПВП стає передбачуваним та не придатним для практичного застосування.

Більшість простих практичних генераторів хоч і мають високу швидкість, але мають багато недоліків:

- занадто короткий період;
- послідовні значення не є незалежними;
- деякі з бітів "менш випадкові", ніж інші;
- нерівномірний однорідний розподіл;
- зворотність.

Наприклад, алгоритм RANDU, що десятиріччями використовувався на мейнфреймах (високопродуктивний комп’ютерзі значним об’ємом оперативної та зовнішньої пам’яті, призначений для централізованих сховищ даних великої ємності та виконання інтенсивних обчислювальних робіт), виявився дуже поганим, що викликало сумніви в достовірності результатівбагатьох досліджень, що використовували цей алгоритм.

Найбільш поширені лінійний конгруентний метод, метод Фібоначі з запізнюванням, регістр зсуву з лінійним зворотним зв’язком.

З сучасних ГПВП широке застосування також отримав"вихор Мерсена", запропонований в 1997 році Мацумотота Нісімурою. Його перевагами є надзвичайно великий період ($2^{19937}-1$), рівномірний розподіл в 623 вимірах (лінійний конгруентний метод дає більш чи менш рівномірний розподіл максимум в 5 вимірах), швидка генерація випадкових чисел (в 2-3 рази швидше, ніж стандартні ГПВП, що використовують лінійний конгруентний метод). Проте, існують алгоритми, що розрізняють послідовність, утворену "вихором Менсена", як невипадкову. Це робить"вихор Менсена"не придатним для криптографії.

Разом з існуючою необхідністю генерувати легко повторювані послідовності випадкових чисел, також існує необхідність генерувати повністю непередбачувані або абсолютно випадкові числа. Такі генератори називаються генераторами випадкових чисел (ГВЧ – англ.randomnumbergenerator, RNG).Оскільки такі генератори частіше за все застосовуються для генерації унікальних симетричних та асиметричних ключів для шифрування, вони найчастіше будуються з комбінації криптостійкого ГПВП та зовнішнього джерела ентропії (саме таку комбінацію заразприйнято розуміти підГВЧ).

Майже всі великі виробники мікрочіпів постачають апаратні ГВЧ з різними джерелами інформації, використовуючи різні методи для їх очищення від неминучої передбачуваності. Проте на даний момент швидкість збирання випадкових чисел усіма існуючими мікрочіпами (декілька тисяч бітів в секунду) не відповідає швидкодії сучасних процесорів.

В сучасних комп’ютерах автори програмних ГВЧ використовують значно швидші "джерела ентропії", такі, як шум звукової картки або лічильник тактів процесора. Інформаційна ентропія – міра хаотичності інформації, невизначеність появи будь-якогосимволу первинного алфавіту. При відсутності інформаційних втрат чисельно дорівнює кількості інформаціїна символ

повідомлення, що передається.До появи можливості зчитування значень лічильника тактів, збирання інформації з "джерел ентропії"було найбільш вразливим місцем ГВЧ. Ця проблема і до сьогодні повністю не вирішена в багатьох пристроях (наприклад, смарт-картках), які таким чином залишаються вразливими. Багато ГВЧ використовують традиційно випробувані, хоч і повільні, методи збору інформації з "джерел ентропії"на зразок вимірювання реакції користувача (рух маніпулятора типу "миша" та ін.), як, наприклад, вPGP(комп'ютерна програма, що дозволяє виконувати операції шифрування (кодування) та цифрового підпису повідомлень, файлів та іншої інформації, представленої в цифровому вигляді)та Yarrow (числовий генератор, що в даний момент не використовується), або взаємодії між потоками, як, наприклад, в Java secure random. Приклади реалізації ГВЧ приведені в табл. 1.1.

## 1.2 ПВБП як ключі шифрування

В усіх сучасних системах шифрування постає проблема генерації ключів шифрування, і в багатьох із них це реалізовано за допомогою ПВБП, в тому числі з використанням регістрів зсуву. Приклади такого використання можна побачити в табл. 1.1

Таблиця 1.1 – Приклади ГВЧ та джерел ентропії

| | Джерело ентропії | ГПВЧ | Переваги | Недоліки |
|---|---|---|---|---|
| /dev/random в UNIX/Linux | Лічильник тактів процесора,проте збирається тільки під час апаратних | LFSR, з хешуванням виходу SHA-1 | Присутній в усіх Unix, надійне джерело ентропії | Дуже довго"нагрівається", може надовго"застрягати", або працює як ГПВЧ(/dev/urandom) |

| | | | | |
|---|---|---|---|---|
| | переривань | | | |
| Yarrow від Брюса Шнайера | Традиційні методи | AES-256 та SHA-1 маленького внутрішнього стану | Гнучкий криптостійкий дизайн | Довго "нагрівається", сильно залежить від криптостійкості обраних алгоритмів, повільний |

Таблиця 1.1 – Продовження

| | | | | |
|---|---|---|---|---|
| Microsoft CryptoAPI | Поточний час, розмір жорсткого диску, розмір вільної пам’яті, номер процесу і NETBIOS-ім’я комп’ютера | MD5-хеш внутрішнього стану розміром 128 біт (хеш присутній тільки в 128-бітових версіях Windows) | Вбудований в Windows, не "застряє" | Сильно залежить від вбудованого криптопровайдера (CSP). |
| Java SecureRandom | Взаємодія між потоками | SHA-1-хеш внутрішнього стану (1024 | В Java іншого вибору поки немає, великий | Повідьне збирання ентропії |

| | | | | |
|---|---|---|---|---|
| | | біт) | внутрішній стан | |
| Chaos від Ruptor | Лічильник тактів процесора, збирається безперервно | Хешування 4096-бітового внутрішнього стану на основі нелінійного варіанту Marsaglia-генератора | Поки самий швидкий з усіх, великий внутрішній стан, не "застряє" | Оригінальна розробка, властивості приведені тільки по затвердженню автора |
| RRAND від Ruptor | Лічильник тактів процесора | Шифрування внутрішнього стану поточним шифром EnRUPT в authenticated encryption режимі (aeRUPT) | Дуже швидкий, внутрішній стан довільного розмірузавибором, не "застряє" | Оригінальна розробка, властивості приведені тільки по затвердженню автора. Шифр EnRUPT не є криптостійким. |

Різновидом ГПВЧ є ГПВБ (PRBG) — генератори псевдо-випадкових бітів, а також потокових шифрів. ГПВЧ, як і поточні шифри, складаються з внутрішнього стану (зазвичай, розміром від 16 біт до декількох мегабайт), функції ініціалізації внутрішнього стануключем (seed), функції оновлення внутрішнього стану та функції виведення. ГПВЧ поділяють на прості

арифметичні, зламані криптографічні та криптостійкі. Їх загальне призначення– генерація послідовностей чисел, які неможливо відрізнити від випадкових обчислювальними методами.

Хоча багато ГПВЧ або потокові шифри пропонують набагато "випадковіші" числа, такі генератори значно повільніші звичайних арифметичних і можуть бути непридатні в різних дослідженнях, що потребують, щоб процесор був вільним для більш корисних обчислень.

У військових цілях та в польових умовах застосовуються тільки засекречені синхронні криптостійкі ГПВЧ (потокові шифри), блокові шифри не використовуються. Прикладами відомих криптостійких ГПВЧ є RC4, ISAAC, SEAL, Snow, зовсім повільний теоретичний алгоритм Блюма, Блюма і Шуба, а також лічильники з криптографічними хеш-функціями або криптостійкими блоковими шифрами замість функції виведення.

Розглянемо деякі приклади криптостійкого шифрування:

*Циклічне шифрування*. В даному випадку використовується спосіб генерації ключа сесії з майстер-ключа. Лічильник з періодом $N$ використовується в якості входу в шифрувальний пристрій. Наприклад, у випадку використання 56-бітового ключа DES використовуватись лічильникз періодом 256. Після кожного створенного ключазначення лічильника збільшується на 1. Таким чином, псевдовипадкова послідовність, отримана по даній схемі, має повний період: кожне вихідне значення$X_0$, $X_1$,…$X_{N-1}$основане на різних значеннях лічильникаі, тому, $X_0 \neq X_1 \neq X_{N-1}$. Так як мастер-ключ є секретним, легко показати, що будь-який секретний ключне залежить від знанняодного або більшепопередніх секретних ключів.

*ANSI X9.17*. ГПВЧ з стандарту ANSI X9.17 використовується в багатьох додаткахфінансової безпекита PGP. В основі цього ГПВЧ лежить потрійний DES. Генератор ANSI X9.17 складається з наступних частин:

Вхід: генератором керують два псевдовипадкових входи. Один є 64-бітним представленнямпоточної дати і часу, які змінюються кожного разупри створенні числа. Інший є 64-бітним початковим значенням. Воно ініціалізуєтьсядеяким довільним значеннямі змінюється в ході генеруванняпослідовності псевдовипадкових чисел.

Ключі: генератор використовує три модулі потрійного DES. Всі три використовують одну і ту ж пару 56-бітних ключів, яка тримається в секретіі застосовується тільки при генеруванні псевдовипадкового числа.

Вихід: вихід складається з 64-бітового псевдовипадкового числа та 64-бітового значення, яке буде використовуватись в якостіпочаткового значенняпри створенні наступного числа.

$DT_i$ –значення дати і часуна початок і-оїстадії генерування.

$V_i$–початкове значення для і-оїстадії генерування.

$R_i$–псевдовипадкове число, створене на і-ійстадії генерування.

$K_1$, $K_2$– ключі, використані на кожній стадії.

Схема включає використання 112-бітового ключата трьох EDE-шифрувань. На вхід подаються два псевдовипадкових значення: значення дати і часуі початкове значення поточної ітерації, на виході отримуємо початкове значеннядля наступної ітераціїі чергове псевдовипадкове значення. Навіть якщо псевдовипадкове число$R_i$буде скомпрометоване, обчислити$V_{i+1}$з$R_i$не представляється можливимза розумний час, та, отже, наступне псевдовипадкове значення$R_{i+1}$, так як для отримання$V_{i+1}$додатково виконуються три операції EDE.

Крім застарілих, добре відомих LFSR-генераторів, широко використовуютьсяв якості апаратних ГПВЧ в XX сторіччі, нажаль, дуже мало відомо про сучасні апаратні ГПВЧ (поточні шифри), оскільки більшість з них розроблено для військових цілей та тримається в секреті. Майже всі існуючі

комерційні апаратні ГПВЧ запатентовані і також тримаються в секреті. Апаратні ГПВЧ обмежені суворими вимогами до витратної пам’яті, тобто "буферної пам’яті", що застосовується генератором для створення послідовності (частіше за все використанняпам’яті заборонене взагалі), швидкодією (1-2 такти) та площею (декілька сотень FPGA- або ASIC-комірок). Через такі суворі вимогидо апаратних ГПВЧ дуже важко створити криптостійкий генератор, тому донині всі відомі апаратні ГПВЧ були зламані. Прикладами таких генераторів є Toyocrypt та LILI-128, що є LFSR-генераторами, та обидва були зламані за допомогою алгебраїчних атак.

Через нестачу хороших апаратних ГПВЧ виробники вимушені застосовувати значно повільніші, але широко відомі блокові шифри (DES, AES) та хеш-функції (SHA-1) в потокових режимах.

Псевдовипадкові послідовності також складають основу технології CDMA (CodeDivisionMultipleAccess) – технології багато станційного доступу з кодовим розділенням каналів. Вони забезпечують розширення спектру і кодове розділення каналів. Розширення спектру проводиться за рахунок модуляції несучого коливання за законом псевдовипадкової послідовності, при цьому використовується прямий метод модуляції (directsequence) і модуляція стрибкоподібними перемиканнями частоти (frequencyhopping). Кодове розділення або розрізнення каналів в системі з CDMA здійснюється за рахунок присвоєння кожному абонентському каналу такої кодової ПВП, яка максимальним чином некорельована з сигнатурними послідовностями інших абонентських каналів. У більшості CDMA система синхронізації між базовими та абонентськими станціями також забезпечується за допомогою псевдовипадкових послідовностей. Це можуть бути як сигнатурні, так і спеціально виділені пілот-сигнальні послідовності з малими значеннями бічних викидів їх автокореляційних функцій.

Серед відомих сімейств ПВП довжини $2^N - 1$ з близькою до ідеальної автокореляцією найбільшого поширення в широкосмуговому зв’язку набули *m*-

послідовності, оскільки генерація цих послідовностей найбільш проста, а їх властивості в порівнянні з іншими вивчені набагато краще. В даний час в світі налічується не одна сотня робіт по $m$-послідовностях, але зацікавленість до них не слабшає. Проте, будучи лінійними, $m$-послідовності характеризуються малим значенням лінійної складності. Даного недоліку позбавлені деякі інші послідовності типу Адамара і, перш за все, послідовності GMW, цікавість до яких значно зросла в останній час. Чисельність сімейства послідовностей GMW при великих значеннях $N$ у багато разів перевищує число $m$-послідовностей. Побудова таких ПВП істотно розширює початкову базу для формування максимальних за обсягом підмножин ПВП з прийнятним рівнем взаємної кореляції, що дозволяє в одних випадках збільшувати число користувачів при заданій завадостійкості, а в інших – знижувати рівень взаємних завад при фіксованому числі користувачів.

Перші системи генераторів послідовностей GMW були побудовані та описані ще в 70-х роках. Їх принцип роботи ґрунтувався на декомпозиційній властивості послідовностей GMW. З цієї причини всі ці генератори отримали назву декомпозиційних. Основна відмінність між ними полягає в кількості форм, що генеруються. Необхідно відмітити, що всі ці генератори характеризуються експоненційним зростанням складності реалізації залежно від $N$ і тому великого практичного розповсюдження вони не одержали. Разом з тим, безперечна корисність декомпозиційних генераторів виявилася в тому, що вони послужили прототипом при створенні одного з найбільш простих генераторів. Так, у 1984 р. Велчем і Шолцем був запропонований метод генерації класів послідовностей GMW, що базується на генерації q-ої m-послідовності. Ця публікація стала початком тріумфального розповсюдження послідовностей GMW і водночас дала могутній імпульс для їх подальшого дослідження. Надалі тими ж авторами було показано, що даний метод може бути поширений на всі класи таких послідовностей. На жаль, через різні причини, піонерські роботи залишилися невідомими.

Подальші десятиріччя не привнесли нічого кардинального в техніку генерації послідовностей. Істотний прорив відбувся тільки в 1997 році, коли на Науково-технічній конференції МТУЗІ був запропонований новий метод генерації двійкових послідовностей GMW, який істотним чином спростив розробку генераторів. Простота даного методу на відміну від методу Велча-Шольца полягає у використанні зсунутих копій двійкової *m*-послідовності і, як наслідок, передбачає використання тільки двійкової логіки. Тому є підстави чекати, що з появою таких генераторів число розробників систем зв'язку, що використовують послідовності GMW збільшиться.

Таким чином, раніше невирішеною частиною загальної проблеми залишається завдання пошуку таких ПВП, значення ВКФ яких при всіх зсувах повинні бути достатньо малими, мати великий період і велику лінійну складність. Рішення цієї проблеми для систем зв'язку з CDMA розширює можливість вибору максимальної за об'ємом множини сигналів із заданою завадостійкістю і полегшує побудову пристроїв синхронізації абонентських приймачів при заданому числі абонентів.

Таким чином, пропонується розпочати вирішення проблеми побудови генераторів послідовностей з дослідження лінійних фільтрів та їх властивостей.

### 1.3.1 Типовий сигнатурний аналізатор

У сфері ймовірнісного тестування важливим є синтез генераторів псевдовипадкових тестових послідовностей. Найбільш часто при формуванні псевдовипадкових послідовностей використовуються два методи. Перший з них, що лежить в основі більшості програмних давачів псевдовипадкових чисел, описується наступним рекурентним співвідношенням

$$X_k = AX_{k-1} + B\left(\mathrm{mod}\ M\right),\ k = 1, 2, 3, \ldots,$$

де $A, B, M$ – постійні числа; $X_0 > 0; A > 0; B \geq 0; M > X_0; M > A; M > B$. Даний метод отримав назву мультиплікативного конгруентного (при $B = 0$) або

змішаного конгруентного (при $B > 0$), а сформована у відповідності з ним послідовність – лінійною конгруентною.

В нинішній час в новій техніці тестування цифрових схем найбільш часто використовується сигнатурний аналіз. Перший сигнатурний аналізатор HP5004A був виготовлений фірмою “Hewlett – Packard”. Призначення цього аналізатора – виявлення помилок в послідовності даних для аналізу, що викликані несправностями контрольованого цифрового пристрою.

Типова структурна схема сигнатурного аналізатора складається з регістра зсуву і суматора по модулю два, на входи якого підключені виходи розрядів регістра у відповідності з твірним поліномом $\varphi(x)$. На рис. 1.1 зображена типова структурна схема сигнатурного аналізатора.

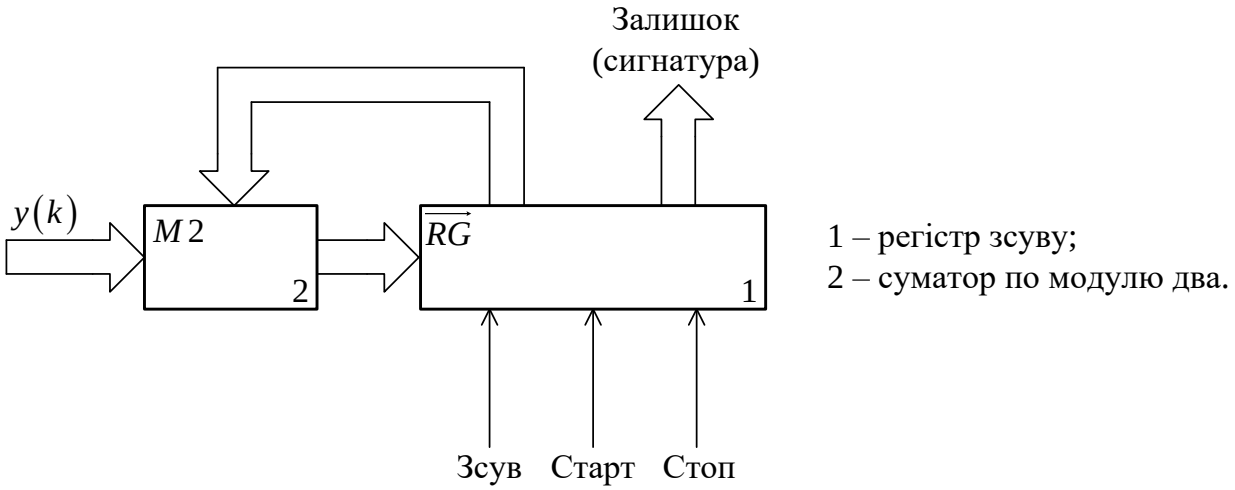



Рисунок 1.1 – Структурна схема сигнатурного аналізатора

Керуючими сигналами сигнатурного аналізатора є "Старт", "Стоп" і "Зсув". Сигнали "Старт" і "Стоп" формують часовий інтервал, протягом якого здійснюється процедура стиснення інформації на аналізаторі. Під дією сигналу "Старт" елементи пам’яті регістру зсуву встановлюються в початкове положення, як правило нульове, а сам регістр починає виконувати функцію зсуву на один розряд вправо під дією синхронізуючих сигналів "Зсув". По

надходженню кожного синхронізуючого імпульсу в перший розряд регістру зсуву записується інформація, що відповідає виразу

$$a_1(k) = y(k) \oplus \sum_{i=1}^{m}{}^{\oplus} \alpha_i a_i(k-1),$$

де $y(k) \in \{0,1\}$ – $k$-й символ послідовності для стиснення $\{y(k)\}$, $k = \overline{1, l}$; $\alpha_i \in \{0,1\}$ – коефіцієнти твірного полінома $\varphi(x)$; $a_i(k-1) \in \{0,1\}$ – вміст $i$-го елемента пам'яті регістра зсуву в $k-1$ такт. Процедура зсуву інформації в регістрі описується співвідношенням

$$a_j(k) = a_{j-1}(k-1),\ j = \overline{2, m}.$$

Таким чином, повний математичний опис функціонування сигнатурного аналізатора має наступний вигляд

$$a_i(0) = 0,\ i = \overline{1, m},$$

$$a_1(k) = y(k) \oplus \sum_{i=1}^{m}{}^{\oplus} \alpha_i a_i(k-1),$$

$$a_j(k) = a_{j-1}(k-1),\ j = \overline{2, m},\ k = \overline{1, l},$$

причому $l$, як правило, приймається рівним або менше величини $2^m - 1$ і, відповідно, визначає довжину послідовності для стиснення. Після $l$ тактів функціонування сигнатурного аналізатора на його елементах пам'яті фіксується двійковий код, що являє собою сигнатуру, що відображується у вигляді шістнадцяткового коду.

Таким чином, шляхом формування тестової послідовності на входах цифрового пристрою для аналізу для кожного його полюсу знаходимо еталонне значення сигнатур, множина яких запам'ятовується і, в подальшому, використовується для порівняння зі значенням сигнатур, що знімаються з пристроїв для перевірки. Будь-яке відхилення реально отриманої сигнатури від еталонної свідчить про те, що полюс схеми функціонує відмінно від випадку

справного стану пристрою. Причина. що могла викликати відмінність сигнатур на даному полюсі, може бути встановлена послідовним аналізом сигнатур від вказаного полюсу до входів пристрою. Дана процедура майже повторює процедуру знаходження несправностей в аналогових пристроях, що полягає в послідовному вимірюванні та аналізі деяких аналогових величин. Такий підхід визначає основну перевагу сигнатурного аналізу: простоту його застосування для визначення і локалізації несправностей цифрових схем, оскільки відсутня складна стендова апаратура при проведенні тестового експерименту і необхідні лише мінімальні навички для його реалізації. Прикладом, що ілюструє простоту технічної реалізації сигнатурного аналізатора, може бути функціональна схема, що описується поліномом $\varphi(x) = 1 \oplus x^7 \oplus x^9 \oplus x^{12} \oplus x^{16}$ (фірма Hewlett – Packard). Даний аналізатор довгий час був неофіційним стандартом пристроїв подібного типу. Його схема складається з суматора з п’ятьма входами по модулю два і 16-розрядного регістра зсуву, а також декількох логічних елементів для організації сигналів "Старт" і "Стоп". Розряди регістра зсуву розбиваються на чотири тетради, вміст кожної з яких визначає значення шістнадцяткової цифри сигнатури по таблиці кодування (табл.1.1).

Таблиця 3.1 – Зв’язок шістнадцяткових значень сигнатур з двійковими

| Двійкове представлення сигнатури | | | | Символ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | A |
| 1 | 0 | 1 | 1 | C |
| 1 | 1 | 0 | 0 | F |
| 1 | 1 | 0 | 1 | H |
| 1 | 1 | 1 | 0 | P |
| 1 | 1 | 1 | 1 | U |

Як конкретний приклад сигнатурного аналізатора розглянемо аналізатор, що реалізований на основі твірного полінома $\varphi(x) = 1 \oplus x^3 \oplus x^4$. Функціональна схема такого аналізатора приведена на рис. 3.2.

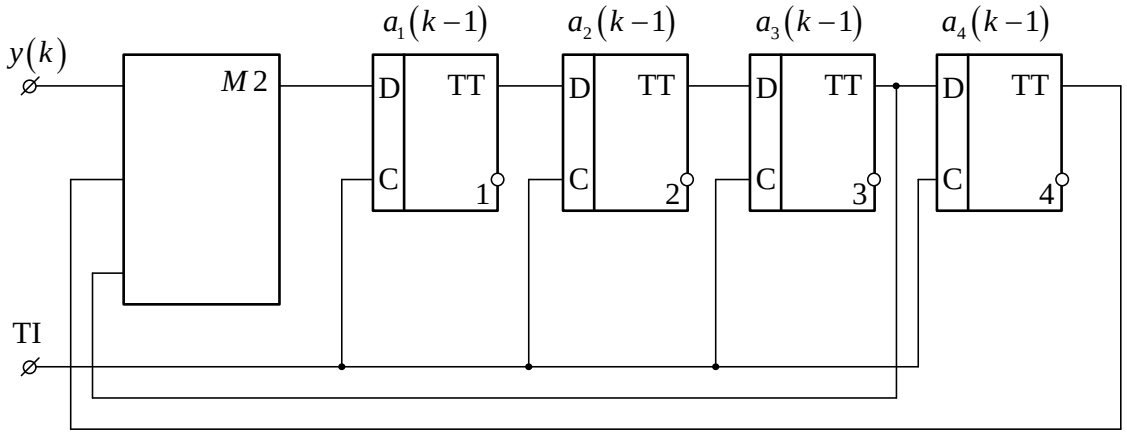



Рисунок 1.2 – Функціональна схема сигнатурного аналізатора

Часова діаграма функціонування такого аналізатора показана на рис. 3.3, де суцільною стрілкою показаний перехід стану $a_1(k-1)a_2(k-1)a_3(k-1)a_4(k-1)$ аналізатора в новий стан $a_1(k)a_2(k)a_3(k)a_4(k)$ при $y(k)=1$, та пунктирною при $y(k)=0$. Ця діаграма якнайкраще показує необхідність та важливість застосування регістрів зсуву в таких аналізаторах.

Для детального розуміння принципів сигнатурного аналізу розглянемо процедуру формування еталонних сигнатур для цифрової схеми, зображеної на рис. 1.4, з використанням аналізатора (рис. 1.2). Припустимо, що для перевірки правильності функціонування схеми використовується комплексне тестування. яке базується на формуванні усіх можливих вхідних наборів у вигляді послідовності лічильників. Еталонні значення реакцій на вхідні набори для кожного полюсу схеми наведені в табл. 1.3.

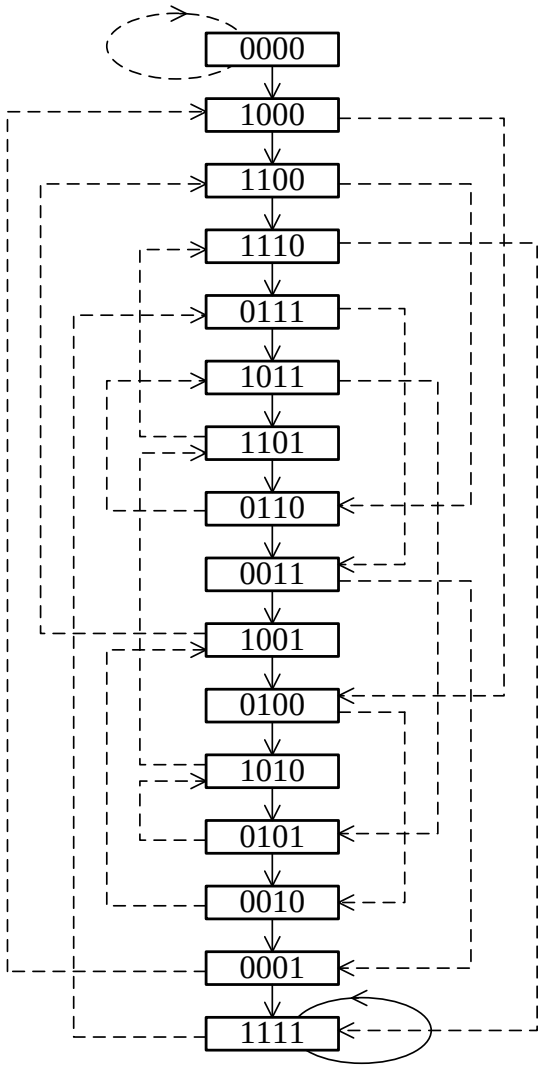



Рисунок 1.3 – Часова діаграма функціонування

сигнатурного аналізатора

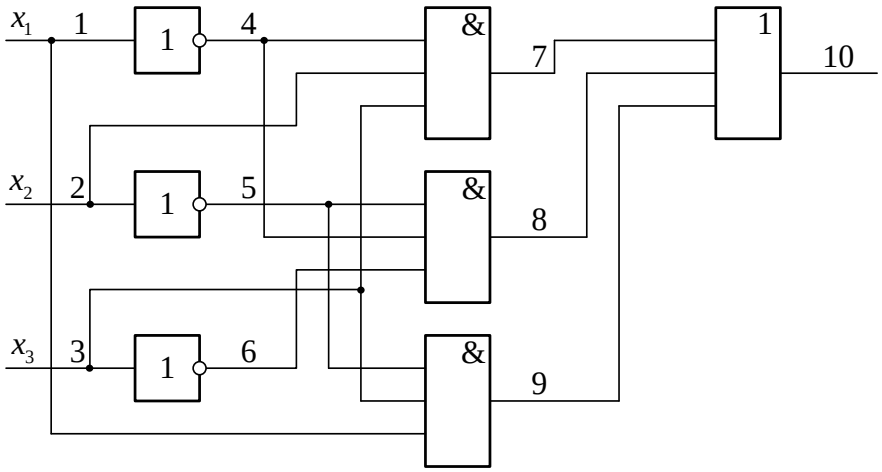



Рисунок 1.4 – Досліджувана цифрова схема

Таблиця 1.2 – Еталонні значення реакцій

| Номер тесту | Тестова послідовність | | | Номер полюсу схеми | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Часова діаграма стану елементів пам’яті аналізатора має вигляд, показаний на рис. 3.5.

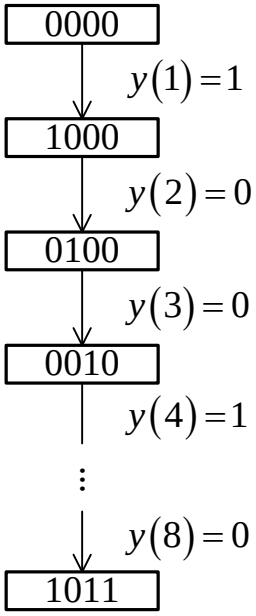



Рисунок 1.5 – Часова діаграма при аналізі послідовності на 10-му полюсі

При аналізі черговий стан аналізатора в залежності від значення $y(k),\ k=\overline{1,8}$, визначався згідно діаграми функціонування аналізатора (рис. 3.3). В результаті врахування останнього переходу для $y(8)=0$ на елементах пам'яті аналізатора формується кінцеве значення сигнатури $S_{10}=1011$. Згідно прийнятого варіанту кодування (табл 3.1), запишемо отриману сигнатуру як $S_{10}=C$. Подібним чином визначаються еталонні сигнатури для інших полюсів. Отримані значення наведені в табл. 3.3.

Таблиця 1.3 – Значення сигнатур для інших полюсів схеми

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $S_i$ | 7 | 6 | $H$ | 4 | 5 | $P$ | $F$ | 5 | 2 | $C$ |

Значення еталонних сигнатур вказуються на принципових схемах цифрового пристрою, подібно до напруг і струмів для аналогових схем, що спрощує організацію процедури виявлення несправності в досліджуваній схемі.

### 1.3.2 Проведення контролю та діагностування несправності схеми

Розглянемо приклад проведення контролю і діагностування несправності схеми, що була наведена на рис. 3.4, в якій через фізичні дефекти на четвертому полюсі зафіксований рівень логічного нуля. Таблиця істинності даної схеми. що містить несправність $f_4 \equiv 0$. містить повну інформацію про її поведінку (рис. 3.4).

Першим етапом при аналізі схеми (рис. 3.4) є визначення реального значення сигнатури на її виході. В результаті стиснення послідовності. що формується на полюсі 10, отримаємо, що $S_{10}^{*} = 2$, а це відповідно є відмінним від еталонного значення $S_{10} = C$. Таким чином, розбіжність сигнатур $S_{10}^{*}$ та $S_{10}$ свідчить про те, що контрольована схема містить несправність. В іншому випадку. коли $S_{10} = S_{10}^{*}$. приймається гіпотеза про те, що контрольована схема знаходиться в справному стані.

Таблиця 3.4 – Таблиця істинності схеми при $f_4 \equiv 0$

| Номер тесту | Тестова послідовність | | | Номер полюсу схеми | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Другий етап при аналізі цифрової схеми – локалізація несправності. Суть її полягає в послідовному аналізі сигнатур на полюсах, функціонально пов’язаних з вихідним полюсом. Дана процедура часто називається процедурою зворотного ходу і для розглянутого прикладу буде полягати в послідовному аналізі сигнатур на полюсах 9, 8, 7. В результаті отримаємо, що $S_9^* = 2, S_8^* = S_7^* = 0$ і відповідно тільки $S_9^*$ дорівнює еталонному значенню. Подальший аналіз показує, що $S_4^* = 4$ відрізняється від його очікуваного значення $S_4 = 4$. В той же час рівність $S_1^* = S_1 = 7$, а також відповідність реальних сигнатур до еталонних для полюсів 2, 3, 5 та 6 дозволяють прийняти рішення про те. що шукана несправність викликана або дефектом першого

елементу "НЕ", або вихідним полюсом. На цьому процедура аналізу цифрової схеми вважається завершеною.

### 1.3.3 Сигнатурний аналіз як спосіб поділу поліномів

Досить поширеним є використання сигнатурного аналізу як способу поділу двійкових поліномів. Цей процес неможливо реалізувати без застосування регістрів зсуву з суматорами по модулю два. Для опису процедури стиснення інформації, основаної на застосуванні примітивних поліномів, використовуються різноманітні математичні моделі і алгоритми. В якості діленого використовується потік даних для стиснення, що описується поліномом $\chi(x)$ степеню $l-1$, де $l$ – кількість бітів в послідовності. Так, наприклад, послідовність 10011 має вигляд полінома $\chi(x)=x^4 \oplus x \oplus 1$. Дільником слугує примітивний поліном $\psi(x)$, в результаті ділення на який отримаємо частку $q(x)$ і залишок $S(x)$, що пов'язані класичним співвідношенням

$$\chi(x)=q(x)\psi(x) \oplus S(x),$$

де залишок $S(x)$, що являє собою поліном степеню, меншого ніж $m=\deg\psi(x)$, називається сигнатурою.

Розглянемо приклад формування сигнатури для потоку данних 11110101, що описується поліномом $\chi(x)=x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus x^2 \oplus 1$, для стиснення якого використовуємо аналізатор, що відповідає поліному $\psi(x)=x^3 \oplus x^2 \oplus 1$ (рис. 3.6).

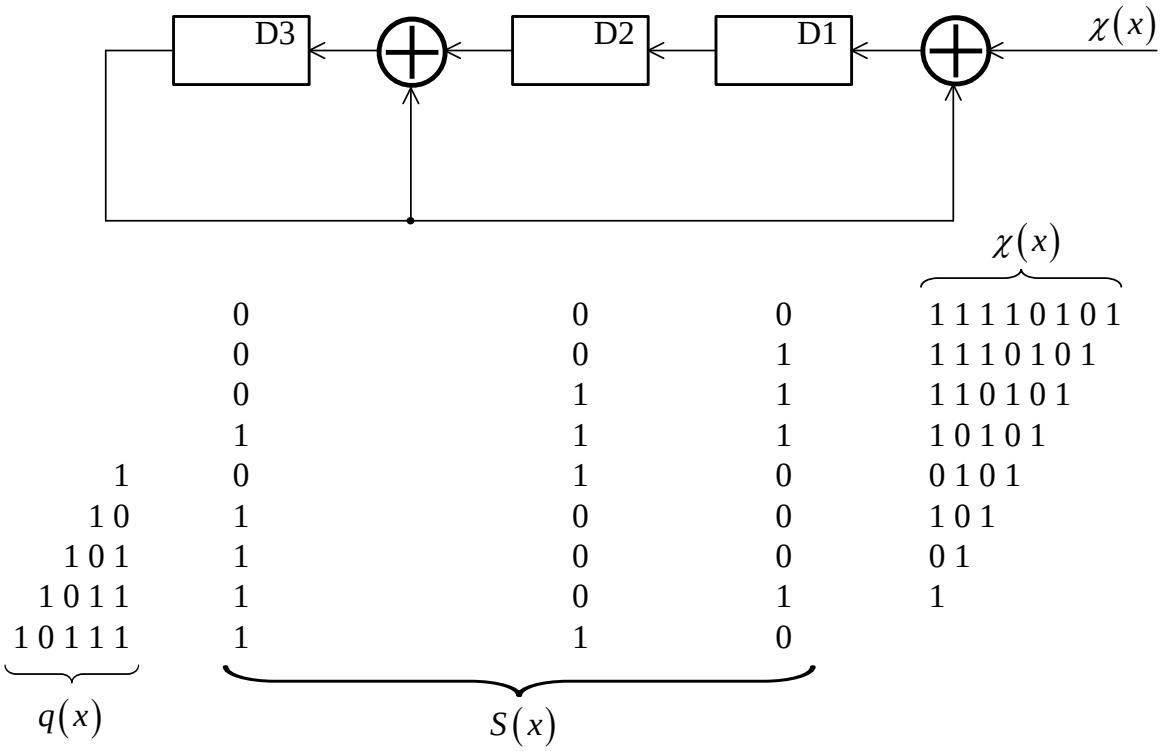



Рисунок 1.6 – Формування сигнатури $S(x)$

Початковий стан елементів пам’яті сигнатурного аналізатора приймається рівним нульовому, хоча в загальному випадку в якості початкового стану може бути будь-який заданий стан. Потік даних для стиснення 1110101 послідовно подається на вхід аналізатора, в результаті чого елементи пам’яті змінюють свій стан по часовій діаграмі, наведеній на рис. 3.6. Залишок $S(x)$ від ділення полінома $\chi(x) = x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus x^2 \oplus 1$ на поліном $\psi(x) = x^3 \oplus x^2 \oplus 1$ фіксується на елементах пам’яті аналізатора і приймає значення $S(x) = x^2 \oplus x$ у вигляді полінома або $S(x) = 110$ – двійкового коду коефіцієнтів полінома. Відповідні значення $S(x)$ залишку від ділення полінома $\chi(x)$ на $\psi(x)$ підтверджуються наступним прикладом:

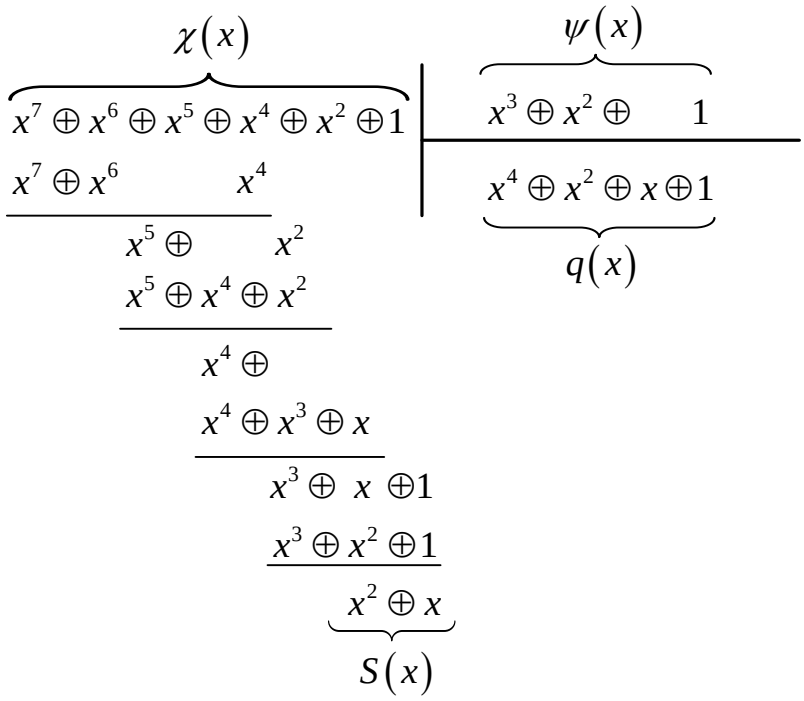



Рисунок 1.7

Звідси слідує рівність залишку від ділення до значення сформованої сигнатури. Доказ даного положення витікає з теорії циклічних кодів. Для практичної реалізації сигнатурного аналізатора, що описується поліномом $\psi(x) = x^3 \oplus x^2 \oplus 1$, існує альтернативна структура, показана на рис. 3.7, яка є кращою з точки зору апаратної побудови. Однак, результат $C(x)$, отриманий при згортці послідовності даних на сигнатурному аналізаторі з зовнішніми суматорами по модулю два, не співпадає з залишком від ділення, тобто $C(x) \neq S(x)$. В той же час між $S(x)$ та $C(x)$ існує однозначна залежність, яка в загальному випадку визначається як

$$S(x) = \begin{vmatrix} \alpha_m & 0 & 0 & \dots & 0 & 0 \\ \alpha_{m-1} & \alpha_m & 0 & \dots & 0 & 0 \\ \alpha_{m-2} & \alpha_{m-1} & \alpha_m & \dots & 0 & 0 \\ . & . & . & . & . & . \\ \alpha_2 & \alpha_3 & \alpha_4 & \dots & \alpha_m & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_{m-1} & \alpha_m \end{vmatrix} \times C(x),$$

де $C(x)$ – результат згортки на сигнатурному аналізаторі, що описується поліномом $\varphi(x)$; $S(x)$ – залишок від ділення полінома $\chi(x)$ на поліном $\psi(x)$, що є зворотним до $\varphi(x)$; $\alpha_i \in \{0,1\}$, $i = \overline{1, m}$, коефіцієнти полінома $\psi(x)$.

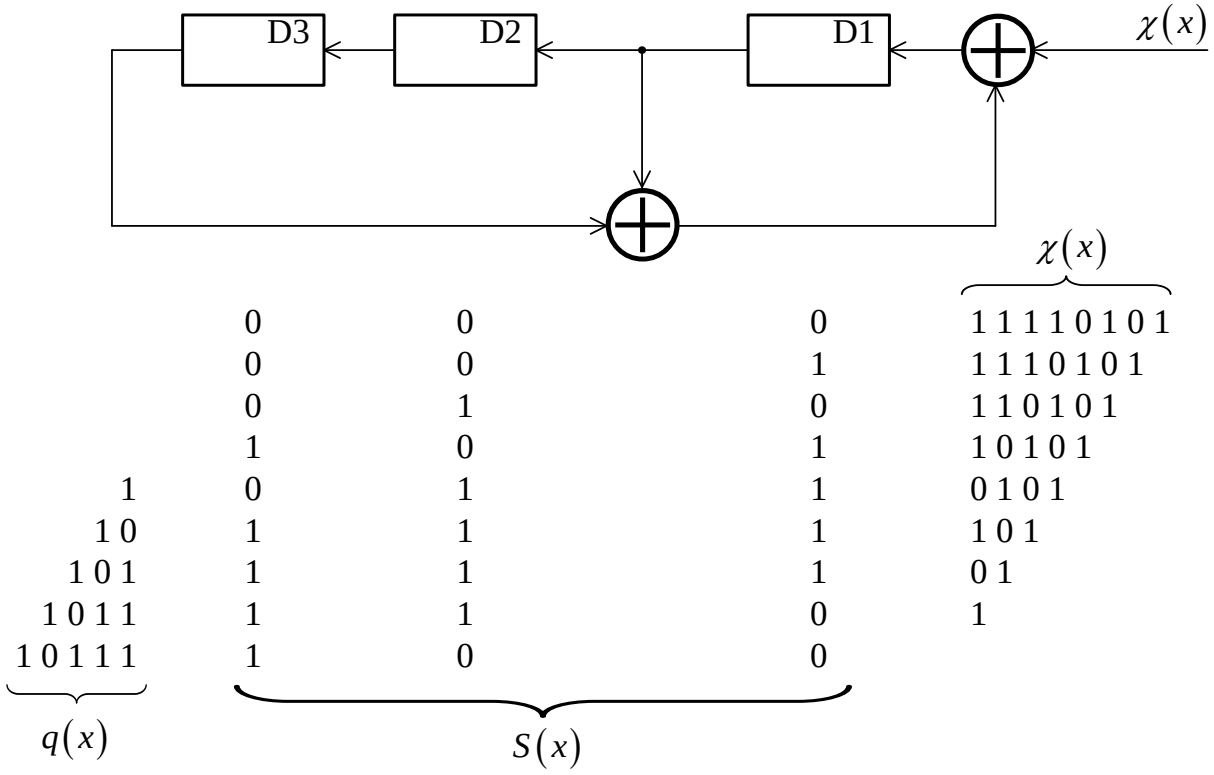



Рисунок 1.8 – Формування сигнатури $C(x)$ для альтернативної структури аналізатора

Лінійність операції додавання по модулю два показує, що описаний вище вираз буде справедливим не тільки для $\chi(x) = x^l$, а й для поліномів $\chi(x)$ будь-якого виду.

Для часткового випадку, представленого на рис. 3.6 та 3.7, сигнатура $S(x)$ утворюється при діленні $\chi(x)$ на поліном $\psi(x) = x^3 \oplus x^2 \oplus 1$, а

значення $C(x)$ – за рахунок стиснення вхідного потоку даних на аналізаторі з зовнішніми суматорами по модулю два, що описується поліномом $\varphi(x) = x^3 \oplus x \oplus 1$, де $\psi(x) = \varphi^{-1}(x)$. Співвідношення для $S(x) = 110$ і $C(x) = 100$ буде мати вигляд

$$S(x) = \begin{vmatrix} \alpha_3 & 0 & 0 \\ \alpha_2 & \alpha_3 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 \end{vmatrix} \times C(x) = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} \times \begin{vmatrix} 1 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 0 \end{vmatrix}.$$

Таким чином, показані приклади сигнатурних аналізаторів і їх математичні моделі дозволяють детально дослідити процедуру формування сигнатур і прослідкувати взаємозв'язок їх теорії синтезу і аналізу з теорією циклічних кодів. Одним з головних критеріїв, що визначили широке застосування сигнатурного аналізу, є його висока достовірність, яка може бути оцінена на основі ряду положень, що витікають з властивостей $M$-послідовностей.

### 1.3.4 Стиснення вихідних реакцій цифрових схем

Доволі важливою сферою використання регістрів зсуву є стиснення вихідних реакцій цифрових схем. Ідея стиснення вихідних реакцій цифрових схем на регістрах зсуву дуже споріднена з основними положеннями сигнатурного аналізу. Структура подібної схеми стиснення складається з регістра зсуву і суматора по модулю два та описується поліномом $\varphi(x) = 1 \oplus x^m$, де $m = \deg \varphi(x)$ – кількість елементів пам'яті регістра зсуву. Функціонування схеми стиснення на регістрі зсуву відповідає системі логічних рівнянь, отриманій для полінома $\varphi(x) = 1 \oplus x^m$

$$a_1(k) = y(k) \oplus a_m(k-1),$$

$$a_j(k) = a_{j-1}(k-1),\ j = \overline{2,\ m},\ k = 1,\ 2,\ 3,\ \ldots,$$

де $a_j(k) \in \{0, 1\}$ – вміст $j$-го елементу пам'яті регістру зсуву в $k$-й такт. Не порушуючи цілісності міркувань, допустимо, що $k = \overline{1,\ l}$, де $l = 2^m - 1$. Тоді послідовно застосовуючи вищеописану систему рівнянь для стиснення вихідної реакції $y(1),\ y(2),\ y(3),\ \ldots,\ y(l)$ через $l$ тактів, отримуємо

$$a_i(l) = \sum_{n=0}^{\lambda}{}^{\oplus}\, y\left(l - mn - i + 1\right) \oplus a_{(m-l_0+i-1)\bmod m}(0),\ i = \overline{1,\ l_0 + 1},$$

$$a_i(l) = \sum_{n=0}^{\lambda-1}{}^{\oplus}\, y\left(l - mn - i + 1\right) \oplus a_{(m-l_0+i-1)\bmod m}(0),\ i = \overline{l_0 + 2,\ m},$$

де $l_0 = l \bmod m;\ \lambda = \left(l - l_0\right) / m$.

Для прикладу $m = 5$ отримаємо, що $l = 2^5 - 1 = 31,\ l_0 = 1$ і $\lambda = 6$. Тоді система рівнянь прийме вигляд

$$a_1(31) = y(31) \oplus y(26) \oplus y(21) \oplus y(16) \oplus y(11) \oplus y(6) \oplus y(1) \oplus a_4(0),$$

$$a_2(31) = y(30) \oplus y(25) \oplus y(20) \oplus y(15) \oplus y(10) \oplus y(5) \oplus y(0) \oplus a_5(0),$$

$$a_3(31) = y(29) \oplus y(24) \oplus y(19) \oplus y(14) \oplus y(9) \oplus y(4) \oplus a_1(0),$$

$$a_4(31) = y(28) \oplus y(23) \oplus y(18) \oplus y(13) \oplus y(8) \oplus y(3) \oplus a_2(0),$$

$$a_5(31) = y(27) \oplus y(22) \oplus y(17) \oplus y(12) \oplus y(7) \oplus y(2) \oplus a_3(0).$$

В загальному випадку початковий стан регістра зсуву може приймати будь-які значення, що не впливає на результат стиснення вихідної

реакції цифрової схеми. Для спрощення припустимо, що $a_1(0)=a_2(0)=a_3(0)=...=a_m(0)=0$, тоді система рівнянь матиме вигляд

$$a_i(l)=\sum_{n=0}^{\lambda}{}^{\oplus} y(l-mn-i+1),\ i=\overline{1,\ l_0+1},$$

$$a_i(l)=\sum_{n=0}^{\lambda-1}{}^{\oplus} y(l-mn-i+1),\ i=\overline{l_0+2,\ m}.$$

В цьому випадку сигнатурою послідовності $y(1),\ y(2), y(3),\ ...,\ y(l)$ буде вміст регістру зсуву через $l$ тактів його роботи:

$$S(y)=a_1(l)a_2(l)a_3(l)...a_m(l).$$

Наведене значення сигнатури $S(y)$ є еталонним для послідовності $\{y(k)\}$, яка в загальному випадку може містити помилкові значення. Будь-яку помилкову послідовність можна описати поліномом помилок або послідовністю $\{z(k)\}$ помилкових значень, причому послідовність $\{z(k)\}$ має таку ж саму розмірність, що і послідовність $\{y(k)\}$, а її символ $z(k),\ k=\overline{1,\ l}$, рівний нулю, якщо реальне значення $y(k)$ приймає вірне значення. і одиниці в іншому випадку. Таким чином, реальна послідовність $\{y^*(k)\}$ подається у вигляді порозрядної суми по модулю два послідовностей $\{y(k)\}$ і $\{z(k)\}$, а сигнатура $S^*(y)$ дорівнює $S(y)\oplus S(z)$. При відсутності помилок в послідовності $\{y(k)\}\ S(z)=000...0$ і відповідно $S^*(y)=S(y)$, а при їх наявності $S(z)$ може приймати довільні значення, причому при $S(z)=000...0$ послідовність помилкових значень бітів, що описуються $\{z(k)\}$, неможливо

визначити. У всіх інших випадках несправності, що характеризуються послідовностями $\{z(k)\}$, для яких $S(z) \neq 000...0$, можна визначити, оскільки

$$S^*(y) = S(y) \oplus S(z) \neq S(y).$$

Це свідчить про те, що реальна сигнатура $S^*(y)$ не відповідає еталонному значенню $S(y)$.

**Висновок**

Оскільки, виходячи зі сказаного вище, актуальність використання ПВБП присутня для таких сфер використання як діагностування електронних схем, шифрування, можна зробити висновок що проблеми їх генерації є актуальними на сьогоднішній день. А саме,синтез псевдовипадкових тестових послідовностей є важливим у сфері ймовірнісного тестування. В нинішній час в новій техніці тестування цифрових схем найбільш часто використовується сигнатурний аналіз. Він слугує способом виявлення помилок в послідовності даних для аналізу, що викликані несправностями контрольного цифрового пристрою. Шляхом формування тестової послідовності на входах цифрового пристрою для аналізу для кожного його полюсу можна знайти еталонне значення сигнатур, множина яких запам’ятовується і, в подальшому, використовується для порівняння зі значенням сигнатур, що знімаються з пристроїв для перевірки. Будь-яке відхилення реально отриманої сигнатури від еталонної свідчить про те, що полюс схеми функціонує відмінно від випадку справного стану пристрою. Дана процедура майже повторює процедуру знаходження несправностей в аналогових пристроях, що полягає в послідовному вимірюванні та аналізі деяких аналогових величин.

# 2 СТАТИСТІЧНІ ХАРАКТЕРИСТИКИ ПВБП, ОЦІНКА ЯКОСТІ ПОСЛІДОВНОСТЕЙ

## 2.1 Процедури тестування ПВБП “на якість”

Псевдовипадкові бінарні послідовності (ПСБП) широко використовуються в телекомунікаційних системах для криптозахисту інформаційного трафіку при потоковому і блочному шифруванні. Для створення ПСПБ застосовують апаратні або програмні генератори псевдовипадкових чисел (ГПСЧ). На сьогодні відомо досить велика кількість реалізацій таких генераторів і виникає проблема об'єктивної оцінки їх якості з точки зору використання генеруються послідовностей в якості ключів при блочному шифруванні або параметрів генератора при Скремблювання. У кожному конкретному випадку для обгрунтованого вибору ГПСЧ проводять тестування ПСБП «на випадковість». При великій довжині ПВБП процес тестування виявляється досить трудомістким по тимчасових витратах, що пов'язано з універсальністю більшості відомих тестів [1,2]. Тому бажано тестування проводити цілеспрямовано (селективно), вибираючи той чи інший тест відповідно до деякими зовнішніми вимогами, обумовленими або областю застосування ПСБП, або алгоритмом її генерації.

Так, методика, запропонована в 1999 г. [1], передбачає використання набору з 16 тестів, кожен з яких орієнтований на виявлення конкретного властивості ПСБП, характерного для дійсно випадкової послідовності. Наприклад, найпростіший з них (Monobit, Block Frequence Cummulative Sums Forward reserve) заснований на тривіальної ідеї підрахунку відносних частот нулів і одиниць в послідовності або її фрагментах. Очевидно, що навіть проста послідовність чергуються нулів і одиниць буде успішно протестована як задовільна. Наступними за складністю є тести типу Runs або Long Runs of Queues, контролюючі довжини, що виникають у досліджуваній послідовності так званих стаціонарних ділянок (що складаються тільки з нулів або тільки

одиниць). Фактично, ці тести (як і попередні) виявляють використання примітивних процедур генерації ПСБП, наприклад, формування послідовності шляхом нарощування її довжини за рахунок повторення стаціонарних фрагментів.

Група тестів, які контролюють періодичність в послідовності (Discrete Fourier Transform, Periodic Templates, Aperiodic Templates) також орієнтована на виявлення спрощених процедур генерації, їх принциповим обмеженням є орієнтація на короткі періоди повторення фрагментів. Тести, які перевіряють «випадковість блукання» (Random Excursions, Random Excursions Variant) є вельми перспективними з теоретичної точки зору. Однак, з іншого боку, способи маскування, щоб забезпечити проходження тестів, досить прості і. в той же час, ефективні.

Тест Linear Complexity виявляє умовну складність послідовності з точки зору її аналітичного опису, але обмежений лише лінійними формами уявлення.

До останньої групи належать тести інформаційного характеру (Universal Statistical Test, Approximate Entropy, Lempel-Ziv Complexity), що базуються на вимірюванні кількості інформації, яка міститься в тестованої ПСБП. Використовується підхід, заснований на вимірюванні ефективності компресії або порівняння частот перекриваються фрагментів з можливостями таких подій для дійсно випадкової послідовності.

В цілому, тести NIST, які отримали на сьогодні найбільше поширення і визнання, є хорошим інструментом для порівняльної оцінки різних ПСБП в деякій умовній системі координат. Однак, строго кажучи, отримані оцінки можна розглядати як в певній мірі суб'єктивні, оскільки вони жорстко прив'язані до вибраного набору тестів. Найбільш перспективним, є інформаційний підхід і тому може бути поставлена задача побудови деякої універсальної процедури обчислення таких характеристик ПСБП, які дозволили б оцінити, наскільки конкретна бітова послідовність близька до дійсно випадкової. При цьому істинно випадкової будемо вважати таку послідовність,

в якій будь-який фрагмент довільної довжини з'являється приблизно з однаковою частотою. Наприклад, уявімо собі, що $2^S$ фрагмент ПСБП спостерігається через деякий уявне «вікно» шириною S біт, і для істинно випадкової послідовності все різновидів фрагмента є рівноімовірними.

У цих припущеннях може бути поставлена наступна задача.

А. Задана конкретна бітова послідовність. $W = (w_1, w_2, ..., w_n)$ Необхідно оцінити кількісно, наскільки ця послідовність близька до дійсно випадковою. Тобто мова йде про спробу об'єктивної оцінки якості конкретної ПСБП.

В інших випадках, наприклад, відносяться до криптоанализу, постановка задачі може бути конкретизована за рахунок деякої додаткової інформації, відомої спочатку криптоаналітику.

В. Як і в попередньому випадку задана конкретна ПСБП. Крім того, відомий загальний алгоритм формування послідовності. Наприклад, імовірно відомо, що генератором ПСБП є регістр зсуву з зворотними зв'язками по модулю 2 (LFSR - linear feedback shift register), а тестування має підтвердити або спростувати цю гіпотезу і виявити (виявити) кореляційні залежності між окремими фрагментами ПСБП. Отриманий результат в цьому випадку може бути використаний для організації відповідної атаки, тобто обчислення конкретних зворотних зв'язків в регістрі генератора і його початкові установки.

(Прийняте припущення про наявність додаткової інформації про ГПСЧ випливає з фундаментального принципу Опост Керкгоффса, яке дуже коротко сформулював К.Шеннон: «Противник знає все, крім ключа».)

Очевидно, при вирішенні другого завдання корисно використати цю додаткову інформацію з тим, щоб зменшити трудомісткість обчислень при тестуванні.

Основний і, в багатьох випадках, доступною є інформація про клас апаратних або програмних засобів, які використовуються для генерації ПСБП.

Найчастіше відомо, наприклад, що в якості генератора ПСБП застосований лінійний фільтр з зворотними зв'язками по модулю 2. У цьому випадку конкретна ПСБП однозначно може бути обчислена криптоаналітиків на основі таких параметрів: довжина регістра d; коефіцієнти многочлена, що задає конкретний вид зворотних зв'язків регістра $b_0, b_1, b_2, ..., b_d$; - стартове слово (комбінація), що задає початковий стан регістра (тобто, по суті, перші d біт ПСБП).

Якщо тестована послідовність дійсно випадкова, то ймовірність появи будь-якої іншої послідовності такої ж довжини дорівнює $\frac{1}{2^n}$. Ентропія джерела таких повідомлень максимальна і $H_{\max} = \log_2 2^n = n$

а при відхиленні від рівномірного розподілу $H_{реал} = -\sum p_i \log p_i$

де - $p_i$ ймовірність появи повідомлення. (В нашому випадку, коли всі повідомлення різновірогідні).

Вимір фактичної ентропії на основі статистичного експерименту для реальних значень n навряд чи можливо. До того ж, в досить типових випадках для тестування пред'явлена лише одна ПСБП. Тому будемо відштовхуватися від деякої гіпотетичної, але насправді легко реалізовується практично процедури. Реальна ПСБП, звичайно ж, не випадкова. Тому. $H_{реал} < H_{\max}$ Знайти точне значення $H_{реал}$ для реальних генераторів не представляється можливим. Навіть при дуже скромних (але реальних) значеннях n = 256 ... 512 підрахувати частоти появи кожної з $2^{256}...2^{512}$ можливих ПСБП нереально за будь-який розумний час.

Тому при тестуванні обмежимо довжину аналізованих фрагментів ПСБП. Почнемо з фрагментів довжини $s = 1$ (один біт). Очевидно, в дійсно випадкової ПСБП $p(0) = p(1) = \frac{1}{2}$. Це легко (і швидко) можна перевірити під час тестування, підрахувавши частоти появи 0 і 1.

. $s = 2$ У цьому випадку також легко підрахувати частоти (ймовірності) появи комбінацій 00, 01, 10, 11. Можна очікувати, що для «гарної» ПСБП ці ймовірності будуть близькі до $\frac{1}{4}$.

s = 3.Частота появи комбінацій 000, 001, ...., 111 знову таки для "гарної» ПСБП будуть близькі до $\frac{1}{8}$. І т. д.

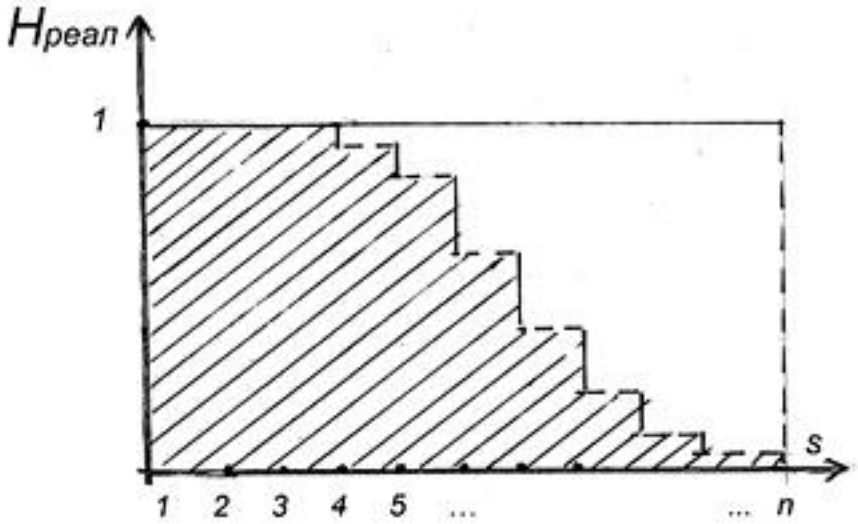
Для підрахунку реальної ентропії скористаємося формулою

$$H_{реал} = -\frac{1}{s}\sum_{i=1}^{q} p_i \log_2 p_i, \quad (1)$$

Де $q = 2^s, s -$ довжина вікна.

В (1) введено множник $\frac{1}{s}$, який дозволяє нормувати отримане в результаті експерименту значення ентропії і «привести» його до одного біту, тобто обчислені таким чином значення не залежать від довжини вікна і стають порівнянними між собою.

Можна впевнено стверджувати, що рано чи пізно статистичний експеримент покаже, що рівномірний розподіл порушується, і $H_{реал} < H_{\max}$. До того ж, трудомісткість підрахунку частот зростає експоненціально і дуже скоро стає обчислювально нездійсненною. Можна припускати, що картина зміни реальної ентропії зі збільшенням довжини фрагмента буде виглядати приблизно так:




Очевидно, в ідеалі (для дійсно випадкової ПСБП) для фрагментів будь-якої довжини ентропія буде залишатися максимальної, і крива її зміни перетвориться в горизонтальну пряму, паралельну осі х-ів. В якості запобіжного якості довільної ПСБП можна використовувати відносну площу заштрихованої області (щодо площі всього прямокутника).

Таким чином, послідовно обчислені ентропії $H_s(s=1,2,...,n)$ повинні добре узгоджуватися (для випадкової послідовності) з теоретичної залежністю, $H_{реал}=H_{\max}$ тобто лінійної. І, навпаки, суттєві відхилення від такої залежності вказують на детермінований характер формування послідовності. При апроксимації такої залежності методом найменших квадратів отримуємо систему

$$a\sum_{i=1}^{k} i^2 + b\sum_{i=1}^{k} i = \sum_{i=1}^{k} iH_i,$$
$$a\sum_{i=1}^{k} i + kb = \sum_{i=1}^{k} H_i.$$

При близькості до вищевказаної залежності необхідно і достатньо, щоб одночасно

$$\tilde{A}(k) = 12\frac{\sum_{i=1}^{k} iH_i - \frac{k+1}{2}\sum_{i=1}^{k} H_i}{ak(k^2-1)} \approx 1,$$

$$\tilde{B}(k) = \sum_{i=1}^{k} iH_i - \frac{k+1}{2}\sum_{i=1}^{k} H_i \approx 0.$$

Відзначимо, що критичним значенням для величини $\tilde{A}(k)$ приймається в даному випадку значення $1-\alpha$, тобто при $\tilde{A}(k)\ A(k) \lhd 1-\alpha$ приймається рішення про детермінованості ПСБП. У той же час значення $k^*$, при якому $A(k^*)$ починає виконуватися ця нерівність, може розглядатися як нижня межа для розрядності передбачуваного генератора тестируемой ПСБП. Аналогічним чином, критичним значенням для величини $\tilde{B}(k)$ є число $\beta \succ 0$, і рішення приймається в разі, коли $B(k) \succ \beta.$

Принципово іншим є підхід, заснований на статистичних оцінках параметрів розподілу кількості нулів і одиниць в тестованої послідовності. Такий підхід може розглядатися як доповнюючий.

Також інформаційним є підхід, заснований на нерівності Крамера-Рао, який обмежує знизу дисперсію оцінки параметра розподілу. У нашому випадку традиційним є підхід, що передбачає рівномірність дискретного розподілу нулів і одиниць в будь-який «вибірці» з тестованої послідовності, що виключає можливість використання згаданого нерівності. У той же час відносна частота $\frac{k}{n}$ (наприклад, одиниць в «вибірці» обсягу n) має асимптотично нормальний розподіл, яке, в свою чергу, допускає використання нерівності. Таким чином, побудувавши для ймовірності p в утвореної послідовністю схемою Бернуллі довірчий інтервал (якому зазначена ймовірність повинна буде належати з ймовірністю $\gamma = 1-\alpha$):

$$(\frac{k}{n} - t_1\sqrt{\frac{k(n-k)(1-\frac{n}{N})}{n^3}}; \frac{k}{n} + t_1\sqrt{\frac{k(n-k)(1-\frac{n}{N})}{n^3}});$$

де - $t_1$ відповідний квантиль нормального розподілу; N - довжина аналізованої послідовності.

Аналогічний інтервал може бути побудований «навколо» апріорі відомої ймовірності $p = \frac{1}{2}$ для, знову таки, апріорі відомої дисперсії емпіричної частоти $\frac{pq}{n} = \frac{1}{4n}$ на основі нерівності Чебишева з використанням нижньої оцінки дисперсії такої оцінки

$$P\left\{\left|D - \frac{1}{4n}\right| \le \varepsilon\right\} \ge 1 - \frac{\overline{D}}{\varepsilon^2} = 1 - \varepsilon,$$

Звідки $\varepsilon_1 = \sqrt{\frac{D}{\varepsilon}}$, де $\overline{D}$ - нижня межа дисперсії по Крамеру-Рао.

Отриманий таким чином довірчий інтервал $\left(\frac{1}{4n} - \varepsilon_1; \frac{1}{4n} + \varepsilon_1\right)$

порівнюється з інтервалом $(\varphi(\alpha); \varphi(\beta))$, де $\varphi = \frac{x(1-x)}{n}, (\alpha \vee \beta) = \frac{k}{n} \pm \sqrt{\frac{k(k-n)(1-n/N)}{n^3}}$.

Що стосується, якщо при прагненні $n \to N$ відношення $\frac{\varphi(\beta) - \varphi(\alpha)}{2\varepsilon_1}$, починаючи з деякого значення n демонструє різке спадання, це може вказувати на штучний характер чергування нулів і одиниць в послідовності, тобто її невипадковий характер.

Також, слід зазначити, що при апаратній реалізації ГПСЧ найчастіше використовуються схеми, які можна віднести до цифрових автоматів Мура тобто таким, у яких вихідні сигнали визначаються внутрішнім станом автомата в поточний момент часу, а перехід до наступного стану відбувається при надходженні вхідного сигналу у вигляді, наприклад, чергового тактового імпульсу. Різноманітність вихідних сигналів в цьому випадку, очевидно, обмежена потужністю безлічі внутрішніх станів автомату, тобто, фактично,

його об'ємом пам'яті. І, найважливіше з точки зору тестування ПСБП, (а це, фактично, послідовність змінюють один одного станів автомата) полягає в тому, що кожен наступний стан однозначно визначається попереднім, і ставиться в цей перехід відповідні автоматні рівняння. При криптоаналізі основне завдання якраз полягає в виявленні пар сусідніх фрагментів, пов'язаних жорсткою функціональною залежністю.

У разі, коли генератор реалізований автоматом Мілі, завдання істотно ускладнюється, оскільки з'являється невідома складова - це вхідні сигнали і зовнішній по відношенню до генератора спосіб їх утворення.

І, нарешті, повертаючись до початкової постановці завдання, підкреслимо, що завдання А відноситься, в основному, до кількісної оцінки якості різних генераторів ПСБП. Очевидно, що для прийняття рішення в цьому випадку про придатність (або непридатність) того чи іншого програмного або апаратного генератора необхідно додатково сформувати пороговий критерій з урахуванням реальних вимог до необхідного рівня захищеності конкретного інформаційного ресурсу.

Завдання В орієнтовано на виявлення наявності кореляційних зв'язків в тестованої послідовності. В інших термінах це завдання зводиться до виявлення закономірностей в процедурі формування ПСБП, які дозволятимуть, у разі їх виявлення передбачити всю послідовність по її фрагменту обмеженої довжини. А це вже одна з основних задач криптоаналізу

## Розділ 2.2 Тести NIST

Тестування генераторів випадкових та псевдовипадкових чисел (ГВЧ і ГПСЧ), використовуваних в криптографічних додатках, є актуальним завданням як у практичному, так і в теоретичному плані. Незважаючи на значні напрацювання в цій галузі, розробники, проте, потребують зручного інструментарію, здатного надати прийнятну метрику, яка дозволить досить

ясно досліджувати ступінь випадковості послідовностей, що породжуються ГВЧ (ГПВЧ), і забезпечити розробників достатнім обсягом інформації для прийняття рішення щодо " якості "генератора.

На сьогоднішній день розроблено досить велику кількість різних типів ГВЧ (ГПВЧ). Однак для демонстрації їх статистичних властивостей використовувалися різні підходи до статистичного тестування. Найчастіше набір і методику тестування пропонував сам розробник генератора. Таким чином, склалася ситуація, яка характеризується тим, що неможливо об'єктивно порівняти різні генератори з єдиних позицій. Виходом з цього положення є використання деякого стандартного набору статистичних тестів, об'єднаних єдиною методикою розрахунку необхідних показників ефективності ГПСЧ і прийняття рішення про випадковість формованих послідовностей. Найбільш відомим набором статистичних тестів є набір з п'яти тестів, запропонований Кнутом в його класичній роботі "Мистецтво програмування для ЕОМ". Вирішенню цього завдання були присвячені ряд робіт вітчизняних авторів. Однак запропоновані рішення мали

- недоліки, які вплинули на їх практичну значимість. Так були запропоновані тільки тести, тоді як питання методики їх застосування розглянуті не досить повно. Однак цим роботам притаманний один недолік

- обмежена кількість статистичних тестів.

У США був зроблений перший крок до стандартизації набору статистичних тестів шляхом прийняття в 1994 році національного стандарту "Вимоги безпеки до криптографічних модулів" . Однак вимоги і методика стандарту носять більше технологічний характер. Вони спрямовані на вирішення завдання статистичного контролю використовуваних в криптографічних модулях псевдовипадкових послідовностей і в загальному випадку малопридатні до вирішення завдання дослідження статистичних властивостей ГПСЧ.

У 1999 році фахівцями NIST, в рамках проекту AES (Advanced Encryption Standard) був розроблений набір статистичних тестів NIST STS (NIST Statistical Test Suite) і запропонована методика проведення статистичного тестування ГСЧ (ГПСЧ) , які на даний момент найкращим чином відповідають потребам всіх зацікавлених сторін.

### 2.2.1 Критерії прийняття рішення про проходження тесту

Для прийняття рішення про проходження послідовністю випадкових (псевдовипадкових) чисел статистичного тесту використовуються наступні три основних обчислювальних підходу.

Нехай дана двійкова послідовність $S = \{s_1, s_2, \ldots, s_n\}$, $s_i \in \{0,1\}$довжиною n біт. Необхідно прийняти рішення, проходить дана послідовність статистичний тест чи ні. Можливі такі підходи до вирішення цього завдання.

1. Критерій прийняття рішення на основі завдання порогового рівня $cR_{пор}R(S)$. . Даний підхід заснований на обчисленні по даній послідовності S статистики тесту c (S) з її подальшим порівнянням з деякими пороговим рівнем cRпорR (S). Критерій прийняття рішення формулюється так: вважається, що двійкова послідовність S не проходить статистичний тест щоразу, коли статистика тесту c (S) приймає значення менше, ніж граничний рівень $cR_{пор}R(S)$.

Наприклад, при перевірці складності послідовності з використанням тесту на основі алгоритму Лемпеля-Зива, по заданій двійковій послідовності S обчислюється її статистика c (S). Для того, щоб визначити, чи пройшла ця послідовність тест чи ні, необхідно порівняти отримане значення c (S) з граничним значенням $n/\log_2 n$. Однак такий підхід не є достатньо надійним. Як показали практичні дослідження використання такого критерію часто призводить до помилкових рішень.

2. Критерій прийняття рішення на основі завдання фіксованого довірчого інтервалу. При цьому підході критерій прийняття рішення формулюється так:

вважається, що двійкова послідовність S не проходить статистичний тест, якщо значення статистики тесту c (S) знаходиться поза межами довірчого інтервалу значень статистики, обчисленого для заданого рівня значущості □. Наприклад, нехай до двійковій послідовності S довжиною n = 800 біт застосовується частотний тест. Значення статистики тесту c (S) є число одиниць в послідовності S, причому очікується, що в послідовності буде приблизно 400 одиниць і 400 нулів. Якщо зафіксувати рівень значущості на рівні 5% (α = 0,05),, то послідовність S не пройде частотний тест, якщо число

одиниць буде знаходитися поза довірчого інтервалу

$$400 \pm 1,96 \times \sqrt{800} = [373,427].$$

Цей критерій є більш надійними, ніж його першим. Необхідно тільки враховувати, що різним рівням значимості будуть відповідати різні довірчі інтервали.

3. Третій підхід побудови критерію прийняття рішення спирається на обчислення для статистики тесту c (S) відповідного значення ймовірності P. Тут статистика тесту розглядається як реалізація випадкової величини, яка підпорядковується відомому закону розподілу. Статистка тесту будується таким чином, щоб її великі значення вказували на будь-якої дефект випадковості послідовності. Значення ймовірності P є ймовірність того, що статистика тесту прийме значення більше, ніж спостерігається на досвіді в припущенні випадковості послідовності. Отже малі значення P (P <0,05 або P <0,01) інтерпретуються як доказ того, що послідовність не випадкова. Вирішальне правило формулюється так: для фіксованого рівня значущості □ двоичная послідовність S не проходить статистичний тест, якщо значення ймовірності P <□. Значення □ рекомендується вибирати з інтервалу [0,001 ÷ 0,01].

Наприклад, нехай послідовність S містить 10^6 біт. Застосуємо до послідовності тест серій, статистикою якого є загальна кількість серій V, яке в

даному випадку має бути близько до значення 500 000. Припустимо, що в ході тестування ми отримали V = 499996. Тоді

$$P = erfc\left(\left|\frac{V - 2n\rho(1-\rho)}{2\sqrt{2n}\,\rho(1-\rho)}\right|\right) = 0{,}99487666,$$

де n - довжина послідовності; ρ - загальна кількість одиниць, поділене на n; erfc - додаткова функція помилок.

Оскільки P> 0,01, то послідовність S тест пройшла.

Використання даного підходу має додаткову перевагу в порівнянні з попереднім, яке полягає в тому, що одного разу розраховане значення ймовірності P може порівнюватися з довільно обраним рівнем значущості  без проведення додаткових розрахунків. Зазвичай значення ймовірності P визначається з використанням

- функції стандартного нормального розподілу

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{\frac{-u^2}{2}}\, du\,;$$

- додаткової функції помилок erfc

$$erfc = \frac{2}{\sqrt{\pi}} \int_{-z}^{\infty} e^{-u^2}\, du\,;$$

- неповною гамма-функції Q (a, x)

$$Q(a,x) \equiv 1 - P(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt \,,$$

де Q (a, 0) = 1 і Q (*a*,∞) = 0.

В основу найбільш потужних бібліотек статистичного тестування ГПСЧ, до яких можна віднести пакет DIEHARD [9], Crypt-SX і пакет NIST STS, закладений третій критерій прийняття рішення.

### 2.2.2 ВПакет NIST STS

Пакет NIST STS включає в себе 16 статистичних тестів, які розроблені для перевірки гіпотези про випадковість довічних послідовностей довільної довжини, що породжуються ГВЧ або ГПВЧ. Всі тести направлені на виявлення різних дефектів випадковості. Основним принципом тестування є перевірка гіпотези, що тестована послідовність є випадковою. Альтернативною гіпотезою є гіпотеза про те, що тестована послідовність не випадкова. За результатами застосування кожного тесту нульова гіпотеза або приймається, або відкидається. Рішення про те, чи буде задана послідовність нулів і одиниць випадкової чи ні, приймається за сукупністю результатів усіх тестів.

Порядок тестування окремої двійковій послідовності S виглядає наступним чином.

1. Було висунуто - припущення про те, що дана двійкова послідовність S

випадкова.

2. По послідовності S обчислюється статистика тесту C (S).

3. З використанням спеціальної функції і статистики тесту обчислюється значення ймовірності

$P = f(c(S))$, $P \in [0, 1]$.

4. Значення ймовірності P порівнюється з рівнем значущості $\alpha$, $\alpha \in [0{,}001;\ 0{,}01]$. Якщо $P \geq \alpha$, то гіпотеза $H$R0 приймається. В іншому випадку приймається альтернативна гіпотеза.

Як вже було сказано, пакет включає в себе 16 статистичних тестів. Але фактично, в залежності від

вхідних параметрів, обчислюється 189 значень ймовірності P, які можна розглядати як результат роботи окремих тестів. У таблиці 1 наводяться зведені дані за всіма тестами із зазначенням кількості обчислюваних значень ймовірності P, фізичного сенсу статистики тесту і дефекту, на виявлення якого спрямований тест (в дужках дан порядковий номер тесту, який використовується на діаграмах).

### 2.2.3 Методика тестування генератора

Розглянутий пакет статистичних тестів може використовуватися для вирішення наступних завдань:

- ідентифікація ГВЧ (ГПВЧ), які формують "погані" виконавчі послідовності;

- розробка нових ГВЧ (ГПВЧ);

- перевірка коректності реалізації ГВЧ (ГПВЧ);

- вивчення генераторів, описаних в стандартах;

- дослідження ступеня випадковості реально використовуваних ГВЧ (ГПВЧ).

При вирішенні перерахованих завдань застосовують таку методику тестування генераторів.

1. З безлічі апаратних або програмних генераторів вибирають генератор G, який необхідно оцінити і прийняти рішення про те, що він формує випадкові виконавчі послідовності. Генератор повинен породжувати двійкову послідовність $S = \{sR1R, sR2R, \ldots, sRnR\}$, $sRiR \in \{0,1\}$, довільної довжини n.

2. Для фіксованого значення n формують безліч з m двійкових послідовностей:

$$S_1 = \{s_1, s_2, \ldots, s_n\};$$
$$S_2 = \{s_1, s_2, \ldots, s_n\};$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$S_m = \{s_1, s_2, \ldots, s_n\}.$$

Таким чином, для тестування необхідно сформувати вибірку обсягом N = m × n.

3. Кожну послідовність піддають тестуванню з використанням пакета NIST STS. В результаті формується статистичний портрет генератора такого вигляду

| № тесту $j$ / № пос-ті $i$ | 1 | 2 | … | $q$ |
|---|---|---|---|---|
| $S_1$ | $P_{1,1}$ | $P_{1,2}$ | | $P_{1,q}$ |
| $S_2$ | $P_{2,1}$ | $P_{2,2}$ | | $P_{2,q}$ |
| : | : | | | |
| $S_m$ | $P_{m,1}$ | $P_{m,2}$ | | $P_{m,q}$ |

$$\begin{pmatrix} P_{11} & P_{12} & \Lambda & P_{1q} \\ P_{21} & P_{22} & \Lambda & P_{1q} \\ M & M & O & M \\ P_{m1} & P_{m2} & \Lambda & P_{mq} \end{pmatrix}$$

Введемо поняття статистичного портрета генератора. Статистичний портрет генератора являє собою матрицю розмірністю $m \times q$, де m - кількість тестованих довічних послідовностей, а q - кількість статистичних тестів, які використовуються для тестування кожної

послідовності. Елементи матриці $P_{ij} \in [0,1]$ де $i = 1, m, j = 1, q$ є значення

ймовірності, отриманої в результаті тестування і-ой послідовності j-им тестом.

4. За отриманим статистичному портрету визначають частку послідовностей, які пройшли кожен статистичний тест. Для цього задають рівень значущості $\alpha \in [0{,}001, 0{,}01]$ і здійснюють

підрахунок значень ймовірності P, що перевищують заданий рівень  для кожного з q тестів, тобто визначають коефіцієнт

$$r_j = \frac{\#\left\{P_{ij} \geq \alpha \middle| i = 1,2,\text{K}\ ,m\right\}}{m}.$$

В результаті формується вектор коефіцієнтів **R** = {*r*R1R, *r*R2R, …, *r*R*q*R}, елементи якого характеризують, в процентному співвідношенні, проходження послідовності SRiR всіх статистичних тестів.

Правило 1. Вважається, що генератор G пройшов тестування по j-му тесту, якщо значення коефіцієнта rRjR знаходиться всередині довірчого інтервалу [*r*R*min*R, *r*R*max*R].

5. Здійснюється статистичний аналіз статистичного портрета. Отримані значення ймовірностей $P_{ij}$ повинні підкорятися рівномірному закону розподілу на інтервалі [0, 1] [2]. Для кожного вектора- стовпчика статистичного портрета будується гістограма частостей $F_k$ влучень значень $P_{ij}$ в кожен з k = 1, 2, ..., 10 підінтервалів, на які розбивається інтервал [0, 1]. Рівномірність розподілу значень ймовірностей $P_{ij}$ перевіряється з використанням критерію $\chi^2$. Для цього обчислюється статистика виду:

$$\chi_j^2 = \sum_{k=1}^{10} \frac{\left(F_k - m/10\right)^2}{m/10},$$

яка підпорядковується розподілу $\chi^2$ с дев'ятьма ступенями свободи.

Правило 2. Вважається, що генератор G пройшов тестування по j-му тесту, якщо виконується умова $\chi_j^2 > 0{,}0001$.

6. Нарешті, приймають остаточне рішення щодо генератора за наступним вирішального правилом: вважається, що генератора G пройшов статистичне тестування пакетом NIST STS, якщо

значення коефіцієнтів $r_j$ для всіх $j = \overline{1,q}$

знаходяться всередині довірчого інтервалу $[r_{min}, r_{max}]$ і

дотримується умова $\chi_j^2 > 0{,}0001$ для всіх $j = \overline{1,q}$

**Висновки**

Можна зазначити, що існує велика кількість реалізацій для ПВБП а тому проблема оцінки випадковості таких послідовностей стоїть гостро, а при великій довжині ПВБП така процедура може займати багато часу, тому існують критерії вибору тестів. Є основні методики тестування, що полягають у підрахунку відносних частот нулів і одиниць в певних фрагментах коду, також методики по виявленню нарощування довжини послідовності, по вимірюванню кількості інформації що міститься в певному блоці послідовності. На даний момент до тестів для ПВБП, що набули найбільшої популярності можна віднести тести NIST. NIST – це стандартний набір тестів, який покликаний дати певну об’єктивну оцінку ПВБП. В цих тестах є два критерії за якими приймається рішення про те чи пройшла ПВБП перевірку, які пов’язані із обчисленням статистики по ПВБП а також з прийняттм рішень на основі фіксованого довірчого інтервалу. Також цей набір тестів може використовуватися для тестування генератора і ідентифікації ГВЧ (ГПВЧ), які

формують "погані" виконавчі послідовності, розробки нових ГВЧ (ГПВЧ), перевірки коректності реалізації ГВЧ (ГПВЧ) і т.д.

# 3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ГЕНЕРАЦІЇ НА ОСНОВІ ВИКОРИСТАННЯ РЕГІСТРОВИХ СТРУКТУР

## 3.1 Класична схема

Отже докладніше зупинимось на генераторі псевдовипадкових чисел що являє собою сукупність $r$ генераторів двійкових послідовностей. При такому підключенні період двійкової послідовності та, відповідно, величина регістрів зсуву генераторів визначається необхідною довжиною багаторозрядної псевдовипадкової послідовності.

Можна запропонувати декілька варіантів реалізації даного генератора. Найпростішим і найпримітивнішим в своїй задумці є генератор з незалежними регістрами зсуву та зворотними зв’язками по модулю два. Він представляє собою набір незалежних логічних кіл, що не чинять будь-якого впливу одне на інше. Максимальний період отриманої послідовності залежить лише від розряду регістра зсуву та вибраного характеристичного поліному. Розрядність регістрів зсуву для кожного такого логічного кола не обов’язково повинна бути однаковою. В цьому полягає особливість даного методу реалізації. На рис. 1.9 наведений конкретний приклад реалізації даного пристрою.

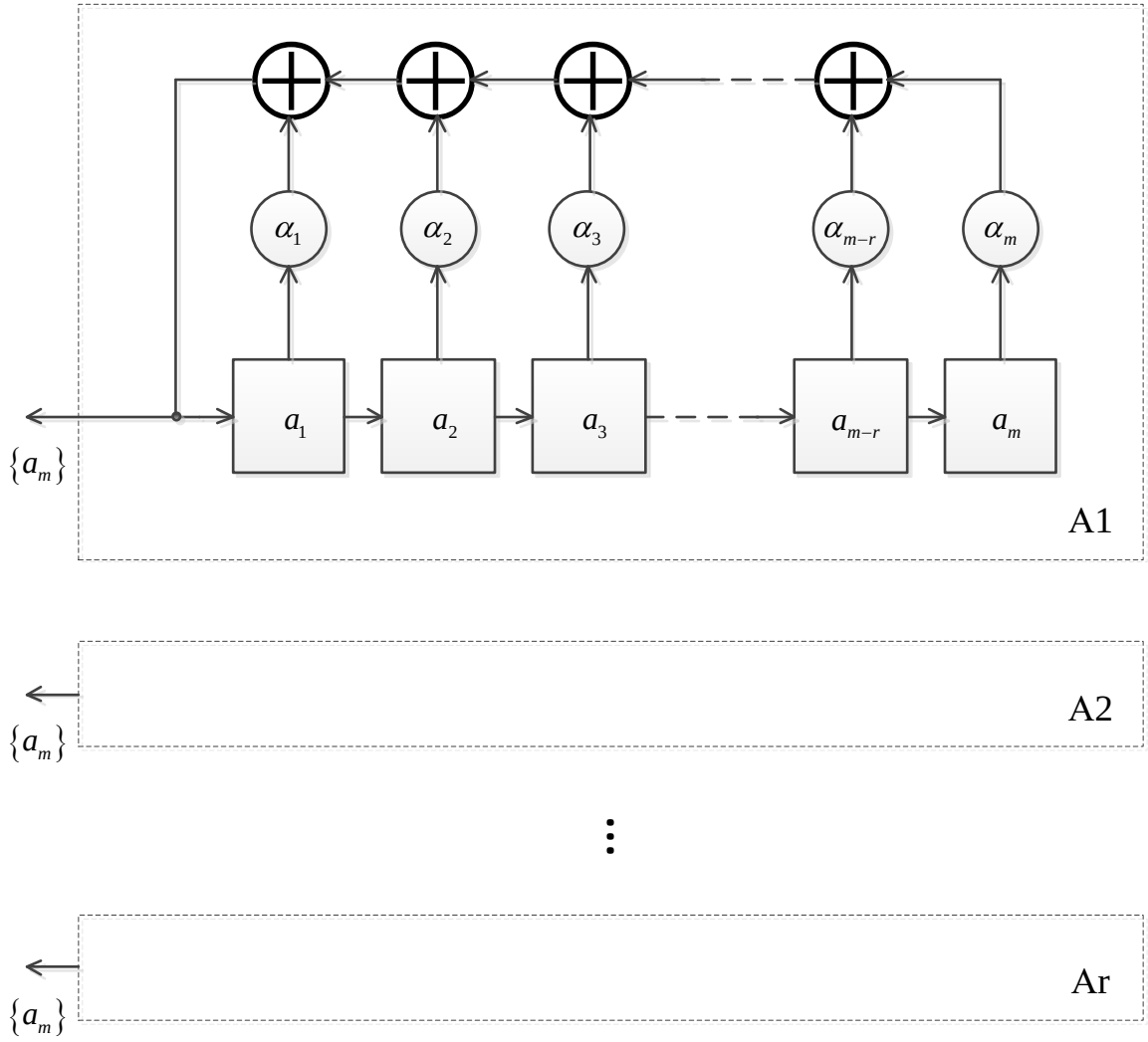



Рисунок 1.9 – Генератор ПВЧП на розділених регістрах зсуву

Більш цікавою є апаратна реалізація генератора з взаємозалежними зворотними зв’язками по модулю два. Така реалізація дозволяє дещо ускладнити отриману послідовність за рахунок взаємозалежних зворотних зв’язків (рис. 5.3)

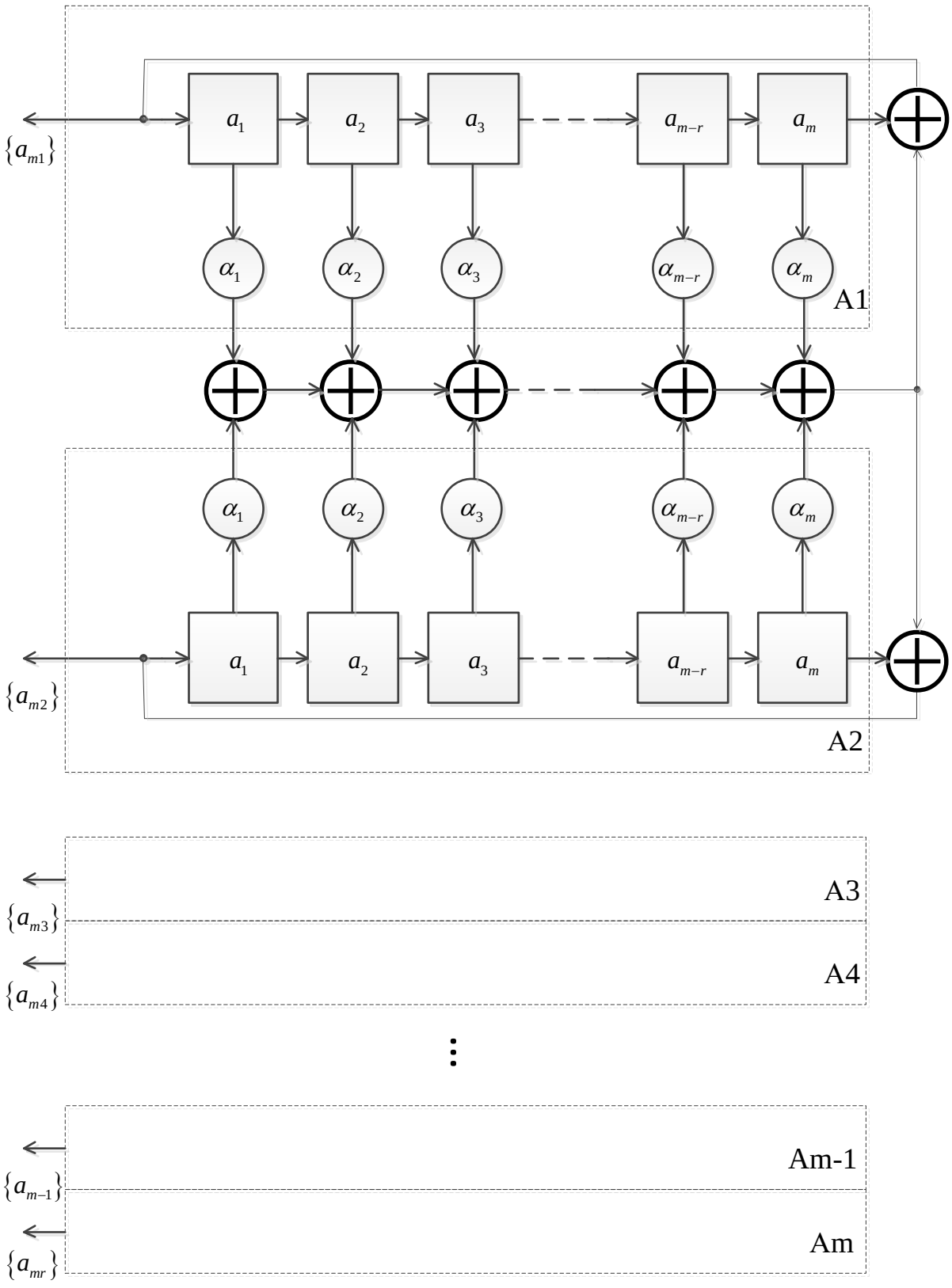



Рисунок 1.10 – Генератор ПВЧП на розподілених регістрах зсуву з взаємозалежними зворотними зв’язками

Не важко зрозуміти, що зі збільшенням необхідної розрядності послідовності буде збільшуватись і фізичний розмір генератора. На практиці подібний підхід для генераторів не отримав поширення, в першу чергу через невисокі статистичні характеристики генерованих послідовностей ПВЧ із-за наявності взаємної кореляції між бінарними послідовностями.

Найчастіше на практиці при побудові $r$-розрядних генераторів ПВЧП використовується так званий послідовний принцип формування ПВЧ [17], що полягає в формуванні чергового значення з $r$ символів, що послідовно отримуються з генератора двійкової послідовності. Структурна схема такого пристрою показана на рис.5.4.

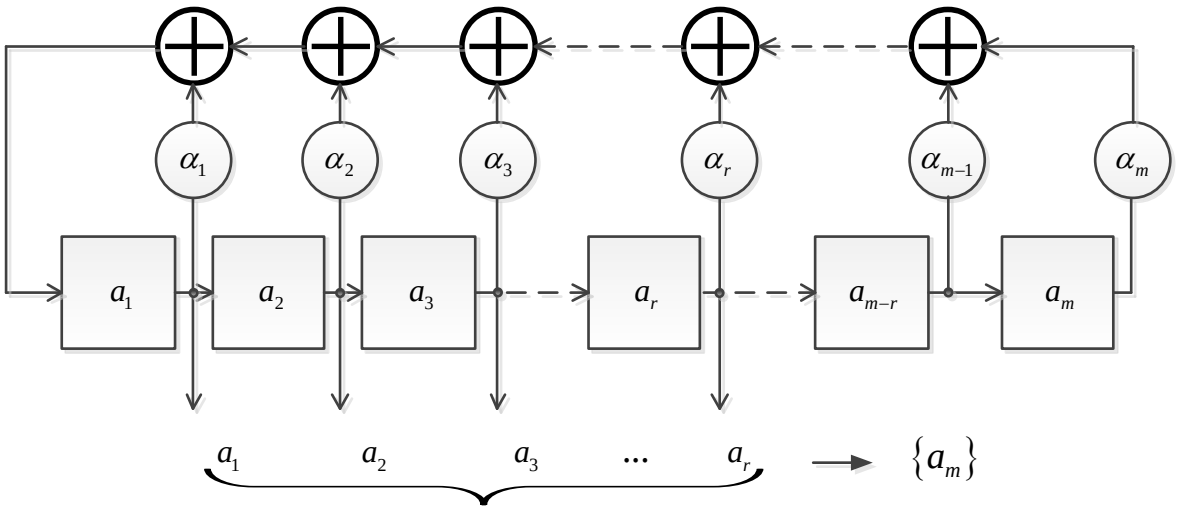



Рисунок 5.4 – Послідовний генератор ПВЧП

Чергове двійкове число $A_k$ утворюється на виходах $r$ розрядів регістра зсуву через кожні $s \geq r$ тактів роботи. Це співвідношення є умовою статистичної незалежності суміжних чисел в формованій послідовності. Довільний член ПВЧП $\{A_k\} = A_0, A_1, A_2, ..., A_k, ...$ може бути описаний виразом

$$A_k = a_1(ks)a_1(ks-1)a_1(ks-2)\ldots a_1(ks-r+1), \quad (5.1)$$

де $a_1(ks)$ – вміст першого розряду регістру зсуву в $ks$-й такт роботи генератора. Враховуючи, що послідовності $\{a_1(ks)a_1(ks-1)a_1(ks-2)\ldots$ $\ldots a_1(ks-r+1)\}$, $k=0,1,2,\ldots$ періодичні, неважко також показати періодичність $\{A_k\}$ з величиною, рівною частці від ділення числа $2^m-1$ ($m$- розрядність регістра зсуву генератора) на найбільший загальний дільник чисел $2^m-1$ та $s-(2^m-1, s)$. Якщо $(2^m-1, s)\neq 1$, то генератор буде формувати $(2^m-1, s)$ різних ПВЧП $\{A_k\}$, вигляд і характеристики яких залежать від початкового стану регістру зсуву. При виборі $s$ взаємно простим з величиною $L=2^m-1$ період $\{A_k\}$ рівний $L$, а її характеристики не залежать від початкового стану регістру зсуву. Таким чином, для правильного вибору величини $s$ необхідно користуватись розкладанням числа $L$ на прості множники. [22]

В якості прикладу розглянемо побудову послідовності ГПВЧ для $r=4$ на основі генератора двійкової послідовності при $\varphi(x)=1\oplus x^2\oplus x^5$. Оскільки $L=2^5-1=31$, то можна брати $s=4$. Схема реалізації даного генератора представлена на рис.5.5.

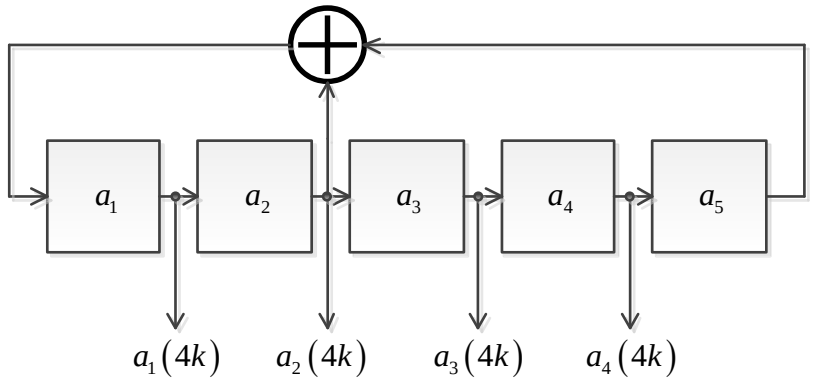



Рисунок 5.5 – Чотирьохрозрядний послідовний ГЧВЧ для $\varphi(x)=1\oplus x^2\oplus x^5$

Послідовності станів регістра зсуву ГПВЧ $\{A_k\} = \{a_1(4k)a_1(4k-1)a_1(4k-2)a_1(4k-3)\}$ при початкових умовах $a_1(0)=1$, $a_i(0)=0$, $i=\overline{2,5}$ приведені в табл.5.1.

Таблиця 5.1 – Послідовність станів елементів генератора ПВЧП і генерованих чисел

| $k$ | $a_1(k)$ | $a_2(k)$ | $a_3(k)$ | $a_4(k)$ | $a_5(k)$ | $a_1(4k)$ | $a_1(4k-1)$ | $a_1(4k-2)$ | $a_1(4k-3)$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | | | | |
| 2 | 1 | 0 | 1 | 0 | 0 | | | | |
| 3 | 0 | 1 | 0 | 1 | 0 | | | | |
| 4 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 5 | 1 | 1 | 0 | 1 | 0 | | | | |
| 6 | 1 | 1 | 1 | 0 | 1 | | | | |
| 7 | 0 | 1 | 1 | 1 | 0 | | | | |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 9 | 1 | 1 | 0 | 1 | 1 | | | | |
| 10 | 0 | 1 | 1 | 0 | 1 | | | | |
| 11 | 0 | 0 | 1 | 1 | 0 | | | | |
| 12 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 13 | 1 | 0 | 0 | 0 | 1 | | | | |
| 14 | 1 | 1 | 0 | 0 | 0 | | | | |
| 15 | 1 | 1 | 1 | 0 | 0 | | | | |
| 16 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Продовження табл. 1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 17 | 1 | 1 | 1 | 1 | 1 | | | | |
| 18 | 0 | 1 | 1 | 1 | 1 | | | | |
| 19 | 0 | 0 | 1 | 1 | 1 | | | | |
| 20 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 21 | 1 | 1 | 0 | 0 | 1 | | | | |
| 22 | 0 | 1 | 1 | 0 | 0 | | | | |
| 23 | 1 | 0 | 1 | 1 | 0 | | | | |
| 24 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 25 | 0 | 0 | 1 | 0 | 1 | | | | |
| 26 | 1 | 0 | 0 | 1 | 0 | | | | |
| 27 | 0 | 1 | 0 | 0 | 1 | | | | |
| 28 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 29 | 0 | 0 | 0 | 1 | 0 | | | | |
| 30 | 0 | 0 | 0 | 0 | 1 | | | | |

Як варіант подібної реалізації, можна використовувати ділянку певної двійкової послідовності для формування незалежних значень розрядів генерованих чисел. Привабливість даної ідеї полягає в її простоті. Додатково використовуючи суматори по модулю два можна отримувати нові ділянки послідовності. Проте реалізація таких генераторів досить проблемна, оскільки вимагає застосування суматорів по модулю два з великою кількістю входів, що саме по собі вимагає великих затрат. Проте потенційною перевагою такої

системи є швидкодія. Одну з можливих концептуальних моделей показано на рис. 5.6.

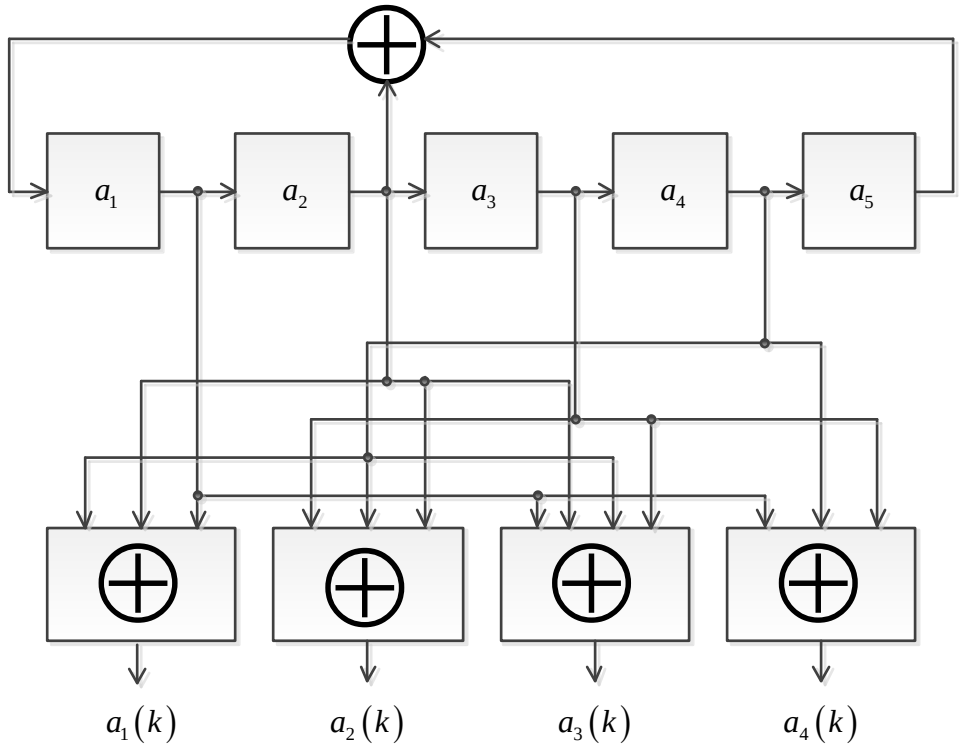



Рисунок 5.6 – Схема реалізація генератора ділянкового типу

### 3.2 Вибір поліномів для зворотніх звязків в загальному випадку

Регістри зсуву з лінійної зворотним зв'язком (Linear Feedback Shift Registers - LFSR) включають власне регістр зсуву і схему обчислення функції зворотного зв'язку (tap sequence) .

Вміст регістра - послідовність бітів - зсувається з приходом тактового імпульсу (clock pulse) на один розряд вправо. Біт наймолодшого розряду вважається виходом LFSR в даному такті роботи. Значення найстаршого розряду при цьому є результатом складання по модулю 2 (функція XOR) розрядів (точок знімання) зворотного зв'язку. Генерируемая послідовність називається лінійної рекуррентой.

Теоретично,n -бітний LFSR може згенерувати псевдослучайную послідовність з періодом $2^n - 1$ біт. Такі LFSR називаються регістрами максимального періоду [1].

Для цього регістр зсуву повинен побувати у всіх $2^n - 1$ ненульових внутрішніх станах.

Одна і та ж рекуррента може бути згенерована регістрами різної довжини. Припустимо, що серед подібних регістрів наш n-бітний LFSR володіє мінімальною довжиною.

Функції зворотного зв'язку регістра можна зіставити поліном m(x) ступені не вище n с коефіцієнтами з поля відрахувань по модулю два, що складається з одночленним виду $x^{n_i - 1}$, де $|n_i|$ безліч номерів точок знімання зворотного зв'язку.

Поліном m(x) називається мінімальним поліномом відповідної рекуррентной послідовності.

Для кожної кінцевої (або періодичної) послідовності можна вказати LFSR, який, при деякому початковому заповненні, породжує цю послідовність.

Серед усіх таких регістрів, існує регістр мінімальної довжини L.

Величина L називается лінейной складністю послідовності S.

Нагадаємо, що поліном називається непріводімим, якщо він не може бути виражений як твір двох поліномів меншій мірі, відмінних констант.

Примітивний поліном ступеня n над полем лишків за модулем два - це не приводиться поліном, який ділить $x^{2^n - 1} - 1$, але не деліт $x^d$ для будь-яких $d : d | 2^{n-1}$.

Теорема. Для того, щоб послідовність, породжена LFSR мала максимальний період, необхідно і достатньо, щоб її мінімальний поліном, був примітивним поліномом по модулю 2.

Список практично застосовних примітивних поліномів наведено в [1,7]. Наприклад, примітивним поліномом є $x^{37}+x^7+x^5+x^3+x^2+x+1$.

Набір показників означає, що, взявши регістр зсуву довжини 32 і генеруючи біт зворотного зв'язку шляхом складання 7-го, 5-го, 3-го, 2-го і 1-го біта по модулю 2, ми отримаємо LFSR максимальної довжини (с $2^{32}$ станами) .

Зауважимо, якщо $p(x)$ - примітивний поліном, то $x^n p(1/x)$- також примітивний. Крім того, якщо поліном $(a,b,0)$прімітівний, то $(a,a-b,0)$- примітивний. Якщо поліном $(a,b,c,d,0)$ прімітівний, то $(a,a-d,a-c,a-b,0)$- примітивний і т.п.

Примітивні тричлени особливо зручні, тому що складаються тільки 2 біти регістра зсуву, але при цьому вони і більш уразливі до атак.

Взагалі кажучи, LFSR - зручні для технічної реалізації, але з точки зору криптографічного стійкості, мають слабкостями.

Послідовні біти лінійної рекурренти лінійно залежні, що робить їх марними для шифрування.

Досить 2n послідовних бітів рекурренти, щоб визначити безліч номерів точок знімання зворотного зв'язку.

### 3.3 Модифікації класичної схеми

Не зупиняючись на теоретичних аспектах синтезу LFSR, зазначимо лише інші важливі сфери їх застосування ПВП в телекомунікаційних системах:

Маскування реальних статистичних характеристик повідомлення ("забілювання") шляхом побітового додавання по модулю 2 ПВП до двійкового повідомлення. Ця процедура суттєво зменшує труднощі тактової синхронізації в мережах цифрової передачі даних та ускладнює криптоаналіз при шифруванні.

Діагностування технічного стану цифрових пристроїв методами сигнатурного аналізу. У цьому випадку до ПВП висуваються досить помірні вимоги: довжина послідовності $2^{20}...2^{30}$ біт. Ці вимоги без проблем задовольняються реалізацією на LFSR.

Потокове шифрування повідомлень шляхом скремблювання. По суті, сама процедура співпадає із "забілюванням", але мета і вимоги до ПВП принципово інші. У цьому випадку довжина послідовності повинна бути якомога більшою, оскільки саме ПВП є ключем шифрування. А "зламати" такий шифр можна лише шляхом підбору ключа або обчисливши параметри генератора ПВП.

Скремблери та дескремблери (шифратори і дешифратори) зазвичай побудовані на основі генераторів псевдовипадкових послідовностей бітів. Генератори частіше за все виконуються з використанням LFSR.

Для того щоб на виході генератора формувалась псевдовипадкова послідовність бітів з періодом повторення, рівним $2^M - 1$, необхідно обирати точки підключення кола зворотного зв’язку у відповідності до степенів примітивних твірних поліномів, що описують ряд генераторів різної розрядності.

Псевдовипадкова послідовність бітів з періодом повторення, рівним $2^M - 1$, має наступні властивості [22]:

В повному циклі ( $2^M - 1$ тактів) число логічних одиниць, що формуються на виході генератора, на одиницю більше. ніж число логічних нулів. Додаткова

логічна одиниця з’являється за рахунок виключення стану, при якому в регістрі був би присутній нульовий код. В результаті ймовірності появи логічного нуля і логічної одиниці на виході генератора практично однакові.

В повному циклі ($2^M - 1$ тактів) половина серій з послідовності логічних одиниць має довжину 1, четверта частина серій – довжину 2, восьма частина – довжину 3 і т.д. Такі ж властивості властиві логічному нулю з урахуванням одного пропущеного логічного нуля. Це свідчить про те, що ймовірності появи 1 та 0 не залежать від попередніх значень. Тому ймовірність того, що серія з послідовних логічних одиниць або нулів закінчиться при наступному кроці складає 0,5.

Якщо послідовність повного циклу ($2^M - 1$ тактів) порівнювати з цією ж послідовністю , але циклічно зсунутою на будь-яке число тактів $W$ ($W$ не є нулем, або числом, кратним $2^M - 1$), то число неспівпадінь буде на одиницю більшим, ніж число співпадінь.

**Розширення періоду генерації**

Самі по собі LFSR є хорошими генераторами псевдовипадкових послідовностей, але вони мають деякі суттєві недоліки. Так, для LFSR довжини $M$ внутрішній стан представляє собою попередні $M$ вихідних бітів генератора. Навіть якщо схема зворотного зв’язку зберігається в секреті, вона може бути визначена по $2M$ вихідним бітам генератора за допомогою деяких високоефективних алгоритмів, наприклад Berlekamp-Massey [12]. Існують також інші методи "зламу" шифрів на основі LFSR шляхом так званих "алгебраїчних атак" [23]. Очевидним шляхом захисту від таких атак є передусім суттєве збільшення довжини ПВП, яка використовується для потокового шифрування.

Використовуючи стандартну схему, збільшення періоду генерації псевдовипадкових послідовностей більше ніж до $2^M$ не представляється можливим. Але можливо запропонувати процедуру, при якій би після

закінчення періоду генерації для одного примітивного полінома, в роботу системи включався б інший. Таким чином для системи певної розрядності період генерації збільшувався б удвічі. Отже можливо створювати системи, в яких період генерації залежав би тільки від кількості примітивних поліномів. Схема такої системи представлена на рис. 6.8.

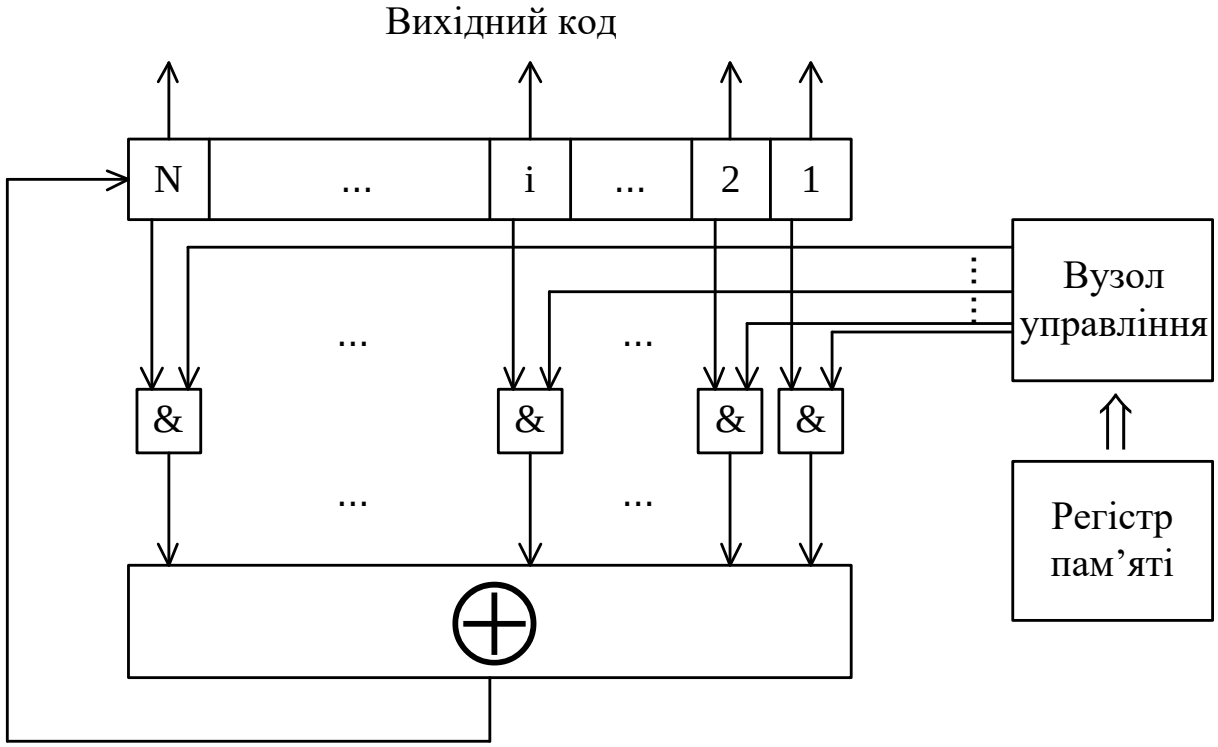



Рисунок6.8 – Генератор псевдовипадкової послідовності з розширенням періоду генерації

В такому генераторі в регістрі пам’яті зберігаються необхідні примітивні поліноми. Вузол управління в свою чергу, в залежності від розрядності системи, вирішує, коли саме "дістати" з регістру пам’яті наступний примітивний поліном і задіяти його в системі.

Корисним є застосування реверсивності в генераторах ПВП [23]. Реверсивне функціонування регістрів зсуву дуже просто реалізується і при цьому не викликає ніяких незручностей. При реверсивному функціонуванні число ПВП може бути збільшене вдвічі. При цьому, якщо зсув інформації у вказаних регістрах відбувається управо, то пристрій генерує одні ПВП, а при зсуві вліво – інші ПВП, "зворотні" до перших.

Функцію вузла управління може брати на себе розподілювач імпульсів [4]. При цьому зникає проблема виявлення твірних поліномів або певних закономірностей в утвореній послідовності. Розподілювач імпульсів використовується разом з комутатором. Комутатор представляє собою набір логічних елементів "АБО" з заздалегідь визначеною логікою. Розподілювач імпульсів, до складу якого також входить регістр зсуву, за рахунок комутатора створює унікальну кодову послідовність. Функцію розподілювача може виконувати мікроконтролер. Послідовність, утворена за рахунок використання такого ключа, має досить низький рівень корельованості розрядів. Викриття структурної схеми генератора ПВП з розподілювачем генератора шляхом аналізу вихідної послідовності представляє собою надзвичайно складну задачу. Структурна схема такого генератора представлена на рис. 6.9.

Пристрій для формування двійкових ПВП на основі такого генератора дозволяє отримати одну або декілька форм ПВП при значно менших технічних затратах в порівнянні з відомими аналогічними пристроями.

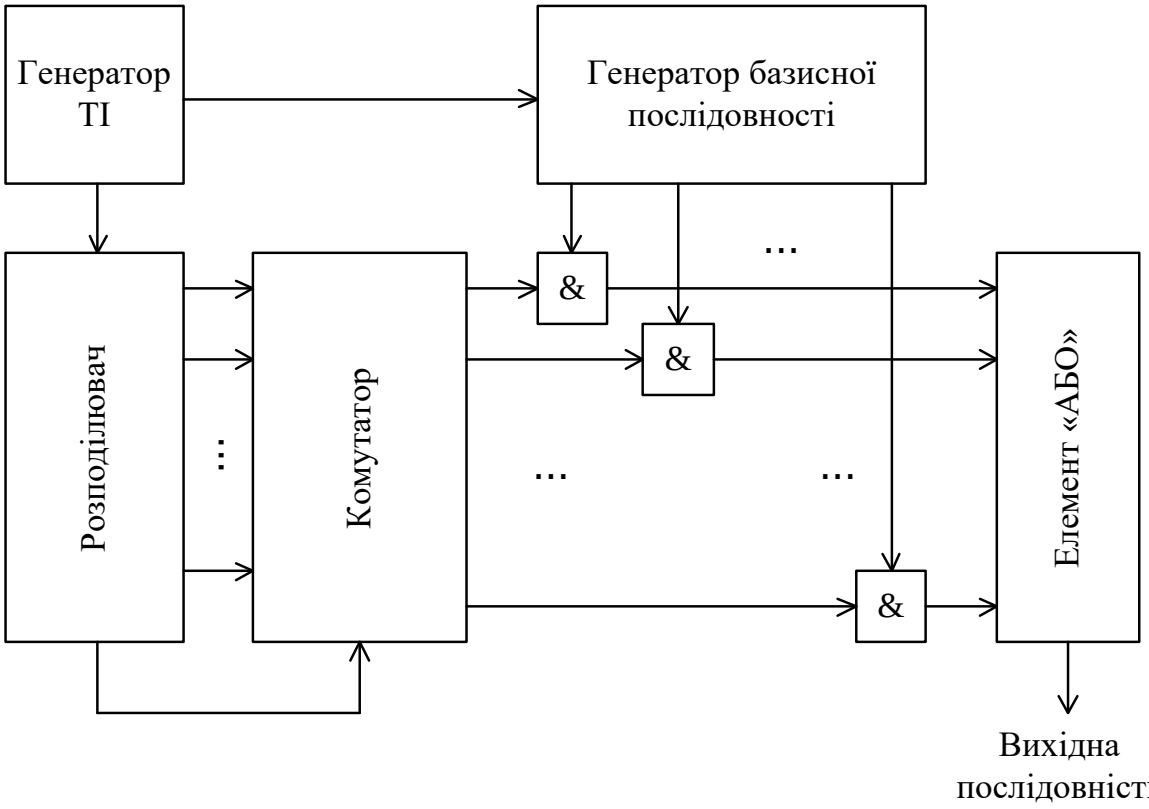



Рисунок6.9 – Структурна схема генератора ПВП з розподілювачем

Ще один варіант генератора ПВП [7, 13] формує псевдовипадкову послідовність бітів з періодом повторювання, рівним $2^M$. Це здійснюється за допомогою додавання нульового стану регістра зсуву. В регістрі $RG$ в певному порядку формуються всі можливі коди, включаючи нульовий. Генератор додатково містить елемент "АБО-НЕ", інвертор і мультиплексор $MS$. Сигнал $Z$ на виході елементу "АБО-НЕ" задає напрям передачі даних через мультиплексор. При $Z=0$ на вихід мультиплексора транслюється сигнал з виходу елемента "Виключне АБО", а при $Z=1$ – сигнал з виходу інвертора. Генератор представлений на рис. 6.10.

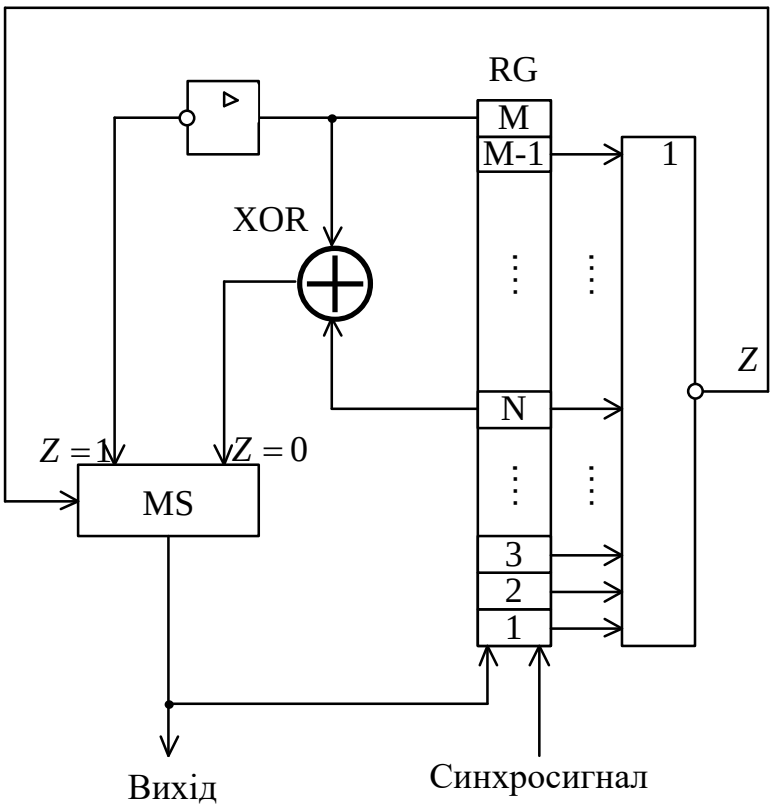



Рисунок6.10 – Генератор ПВП з періодом повторювання $2^M$

До моменту, доки на входах елементу "АБО-НЕ" присутня хоч одна логічна одиниця. на його виході буде сигнал $Z=0$. В цьому випадку мультиплексор $MS$ в кожному такті передає в звільнений (нижній) розряд регістра зсуву біт з виходу елемента "Виключне АБО".

В деякому такті $i$ в регістрі фіксується код, що містить єдину логічну одиницю, розміщену в розряді $M-1$. Оскільки в розрядах $M$ і $N$ присутні логічні нулі то на виході елемента "Виключне АБО" сформований сигнал логічного нуля. який до початку такту $i+1$ надходить на вхід регістра. На початку такту $i+1$ логічна одиниця переміщується з розряду $M-1$ в розряд $M$, на входах "АБО-НЕ" формується нульовий код. Сигнал $Z=1$ переводить мультиплексор $MS$ в стан, при якому на вхід нижнього розряду регістра зсуву надходить біт з виходу інвертора. В даному випадку цей біт рівний нулю, тому в такті $i+2$ в регістрі фіксується нульовий код.

До початку такту $i+3$ на вхід регістра зсуву з виходу інвертора надходить логічна одиниця, тому по фронту синхросигналу в регістрі фіксується код, що містить логічні нулі в усіх розрядах, окрім першого. Сигнал $Z$ знову приймає нульове значення, мультиплексор переключається в стан передачі сигналу з виходу елемента "Виключне АБО" і т.д. Таким чином регістр проходить через усі стани, включаючи нульовий стан.

На завершення розглянемо схему (рис. 6.11) з двома LFSR $R_1$ та $R_2$, де один ($R_1$) працює як генератор ПВП, а другий ($R_1$) змінює кожну згенеровану двійкову комбінацію шляхом фільтрації через сукупність двоходових логічних схем $f_1, f_2, ..., f_i, ..., f_M$. В залежності від виду конкретних функцій $f_i = f_i(x_i, y_i),\ i = 1, 2, ..., M$ та конкретної послідовності, яка створюється $R_2$, результуюча ПВП може змінюватися в широких межах.

Для конкретного набору $f_1, f_2, ..., f_M$ запропонована схема може генерувати ПВП довжиною $2^{3M}$. Для цього блок керування узгоджує темп генерації $R_2$ таким чином, щоб по завершенні кожного циклу $R_1$, змінювався стан $R_1$, а по завершенні повного циклу $R_2$, здійснювалася би заміна стартової комбінації в $R_2$. Щодо вибору фільтруючих функцій $f_1, f_2, ..., f_M$, то це питання досить складне, оскільки вимагає детального аналізу впливу вибраної сукупності функцій на статистичні характеристики результуючої ПВП. Із інтуїтивних міркувань придатними до застосування в пропонованій схемі можуть виявитися насамперед лінійні функції

$$z_i = f(x_i, y_i) = x_i \oplus y_i.$$

Інші булеві функції двох змінних потребують аналізу отриманих ПВП "на випадковість". По суті, мова піде про перевірку появи в ПВП легко прогнозованих закономірностей.

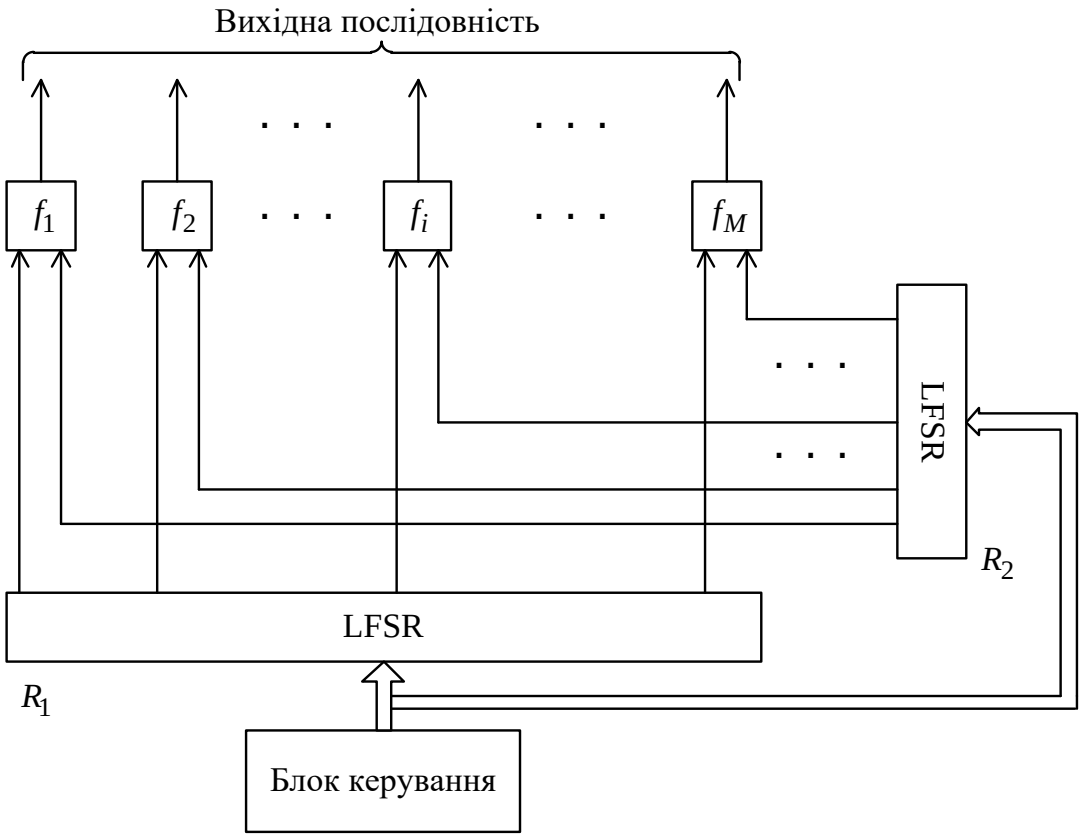



Рисунок6.11 – Генератор ПВП з двома LFSR

Зазначимо також, що у будь-якому випадку пропоновані варіанти нових процедур генерації ПВП дають можливість суттєво ускладнити криптоаналіз як за рахунок збільшення довжини циклу ПВП, так і комбінаторного різноманіття саме процедур генерації. Можна сподіватися, що застосування нових алгоритмів генерації ПВП дозволить ефективно протистояти атакам на потокові шифри класу скремблерів.

**Висновки**

Отже, в класичній схемі реалізації ПВБП період послідовності і величина регістру зсуву визначається довжиною багато розрядної ПВП. Тому варіантами реалізації є генератори з незалежними регістрами зсуву та зв’язками по модулю 2. При такій реалізації максимальний період отриманої послідовності визначається лише регістром зсуву та обраним характеристичним поліномом.

Також застосовуються регістри з лінійним зворотнім зв’язком. В такому регістрі вміст регістру зсувається вправо після приходу тактового імпульсу на вхід, і тому значенням найстаршого розряду стає результат складання по модулю 2 розрдів зворотнього зв’язку. Тому n- бітний LFSR може генерувати ПВБП з періодом $2^n - 1$ біт. Але оскільки існують певні методи зламу таких шифрів на основі LFSR, то є необхідність збільшення періоду гнерації ПВП. Одним з варіантів рішення цієї проблеми є створення систем, в яких період генерації залежав би тільки від кількості примітивних поліномів

# 4 УЗАГАЛЬНЕНИЙ ПІДХІД НА ОСНОВІ МОДЕЛІ ЦИФРОВИХ АВТОМАТІВ

## 4.1 Вибір поліномів для зворотних зв’язків при реалізації

Для реалізації LFSR раніше використовувалися примітивні поліноми над полем Галуа $GF$ (2) з високими ступенями і невеликою вагою Хеммінга (3,5). Вони мають високу продуктивність і мінімальні витрати на апаратну реалізацію (з точки зору кількості логічних елементів XOR, необхідних для побудови LFSR). У той же час вони мають певні недоліки, пов'язані з провалом деяких статистичних тестів , і низькою дифузійної ємкостью. Якісний ГПВЧ повинен

мати характеристичні поліноми, у яких число ненульових коефіцієнтів приблизно дорівнює n / 2, де n - ступінь полінома .

З іншого боку, вибираючи поліноми з великою кількістю ненульових коефіцієнтів, ми приходимо до підвищення витрат на апаратну реалізацію (збільшення кількості логічних елементів XOR). Тут ми повинні шукати компромісне рішення.

Існує досить гарне рішення даної проблеми. Цей метод полягає в пошуку примітивного полінома наступним чином. Якщо функція $f$ $(x)$ може бути представлена у вигляді виразу (1)

$f\ x = 1 + x\ b1\ 1 + x\ b2 \cdots 1 + x\ bm + x\ n$ ,

то LFSR з мінімальним енергоспоживанням, що реалізовується тією ж

самої функцією, може бути побудований за допомогою $m$ елементів

XOR, де

$b1 \geq 1$, $b1 < b2$ , $b1 + b2 < b3$ , ⋯,

$(b1 + b2 + \cdots + bm{-}1) < bm$ , $(b1 + b2 + \cdots + bm\ ) <$ n

Для побудови ефективного ГПВЧ нами були обрано ступінь поліному (89, ), а також значення $m$ = 5. Для пошуку зна-

чень $b1 + b2 + \cdots + bm$ і побудови примітивних поліномів був

використаний пакет Mathematica, за допомогою якого знайдений наступний поліном:

$$P1(x) = (1+x)(1+x^5)(1+x^{10})(1+x^{17})(1+x^{39})+x^{89} =$$

$$x^{89}+x^{72}+x^{71}+x^{67}+x^{66}+x^{62}+x^{61}+x^{57}+x^{56}+x^{55}+$$
$$x^{54}+x^{50}+x^{49}+x^{45}+x^{44}+x^{40}+x^{39}+x^{33}+x^{32}+x^{28}+x^{27}+$$
$$x^{23}+x^{22}+x^{18}+x^{17}+x^{16}+x^{15}+x^{11}+x^{10}+x^{6}+x^{5}+x+1$$

### 4.2 Оцінки складності криптоаналізу при використанні автоматних моделей

Ступінь многочлена задає довжину регістра, ненульові члени описують які елементи регістра складають відвідну послідовність.

Якщо многочлен утворює відповідну послідовність є непривідним по модулю 2, тоді період генерується регістром послідовності буде максимальним і обчислюється за формулою $2^n - 1$.

Перед початком роботи в регістр заноситься довільна послідовність біт, яка називається початковим станом. Після чого кожен такт генератора повертає 1 біт, що виглядає абсолютно випадковим.

Самі по собі РСЛОС є хорошими ГПВЧ, але в силу того, що отримані з їх допомогою біти мають лінійний зв'язок використовувати РСЛОС в криптографічних цілях нерозумно.

Якщо зловмисник отримає послідовність біт довжиною n, що згенерувала за допомогою РСЛОС він може завантажити ці біти в регістр і прокрутивши його назад отримає початковий стан. Знання початкового стану дасть йому

доступ до всіх згенерованих раннє і які згенеруються в майбутньому послідовностям.

Можливо приховаті інформацію про регістр. Тоді, навіть отримавши послідовність довжиною n атакуючий не зможе зробити кроків щодо розкриття початкового стану.

Але така ситуація легко вирішується за допомогою алгоритму Berlekamp-Massey. Даний алгоритм дозволяє розкрити пов'язаний з РСЛОС многочлен. Для цього достатньо мати сгенерированную регістром послідовність довжиною всього 2n.

Алгоритм досить простий в реалізації:

```
public int[] BerlekampMassey(int[] array)
    {
        int N = array.Length;
        int[] b = new int[N];
        int[] c = new int[N];
        int[] t = new int[N];
        b[0] = 1;
        c[0] = 1;
        int l = 0;
        int m = -1;
        for (int n = 0; n < N; n++)
        {
            int d = 0;
            for (int i = 0; i <= l; i++)
            {
                d ^= c[i] * array[n - i];
            }
```

```
            if (d == 1)
            {
                Array.Copy(c, 0, t, 0, N);
                int N_M = (n-m);
                for (int j = 0; j < N - N_M; j++)
                {
                    c[N_M + j] ^= b[j];
                }
                if (l <= n / 2)
                {
                    l = n + 1 - l;
                    m = n;
                    Array.Copy(t, 0, b, 0, N);
                }
            }
        }
        return c;
    }
```

На вхід надходить послідовність біт, згенерована за допомогою РСЛОС. Як результат повертається многочлен, що характеризує схему зворотного зв'язку.

Зрозуміло, РСЛОС можна об'єднувати в каскади для більш криптостійкого ГПСЧ. Ця ідея використовується в деяких поточних шифрах. Однак, багато генераторів, засновані на цьому способі уразливі до так званих кореляційних атак. За допомогою кореляційної атаки, зловмисник володіючи послідовністю згенерованої ГПСЧ має можливість відновити початкове значення і отримати доступ до всіх генеруються в майбутньому значенням.

### 4.3 Програмна реалізація

Реалізація алгоритму на С

```
    int LFSR (void)
    {
      static unsigned long S = 1;
       S  =  (((  (S>>88)^(S>>71)^(S>>66)^(S>>65)^(S>>61)^
(S>>60)^(S>>56)^(S>>55)^(S>>54)^
(S>>53)^(S>>49)^(S>>48)^(S>>44)^(S>>43)^
(S>>39)^(S>>38)^(S>>32)^(S>>31)^
(S>>27)^(S>>26)^(S>>22)^(S>>21)^(S>>17)^
(S>>16)^(S>>15)^(S>>14)^(S>>10)^  (S>>9)^(S>>5)^(S>>4)^S  )
& 1 ) << 88 ) | (S>>1);
      return S & 1;
    }
```

Варто відзначити, що вага Хеммінга кожного з цих поліномів $W = 33$. Така вага є досить хорошою, а для реалізації зазначеного поліному потрібно тільки 5 елементів XOR.

### Висновки

При реалізації ГПВЧ  існують такі проблеми: з одного боку якісний ГПВЧ повинен мати характеристичні поліноми в яких число ненульових коефіцієнтів повинно бути n/2 , а з іншої сторони при виборі поліномів із великою кількістю ненульових коефіцієнтів збільшуються витрати при реалізації такого ГПВЧ.

РЗЛЗЗ є хорошими ГПВЧ, але отримані з їх допомогою біти мають лінійний зв'язок..

Знання початкового стану дасть доступ до всіх згенерованих попередньо і всіх послідовностей, що будуть згенеровані в майбутньому.

Тому було запропоновано оптимальний варіант підбору поліному, та програмної реалізації такого ГПВЧ.

# 5 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

## 5.1 Опис ідеї проекту

Сутність стартап-проекту. Під час дослідження ринку рішень для шифрування, було виявлено можливість впровадження нової реалізації ГПВЧ . Зміст ідеї стартапу та визначення її характеристик наведено в табл. 5.1 та табл. 5.2.

Таблиця 5.1 –Опис ідеї стартап-проекту

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|---|---|---|
| Генератор ПВБП зі зворотнім зв’язком по модулю два із новим поліномом зворотнього звязку | 1. Шифрування інформації | Менші ресурси для реалізації при достатно високій надійності. |
| | 2. Тестування електронних пристроїв. | Швидкий та надійний спосіб виявлення несправностей. |

Опис до таблиці 5.2:

W – слабка сторона;

N – нейтральна сторона;

S – сильна сторона.

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| № п/п | Техніко-економічні характе-ристики ідеї | (потенційні) товари/концепції конкурентів | | | | W | N | | S s |
|---|---|---|---|---|---|---|---|---|---|
| | | ГПВП з незалежними регістрами зсуву | Blowfish | DES | AES | | | | |
| | Трафік за оплатою | + | - | + | - | | | | |
| | Технології | GSM, EDGE, | IP | DVB-S | IP | | | | |
| | Час доступу | + | + | + | - | | | | |
| | Об’єкти прийому | телефон | ПК | SAT-ресивер | ПК | | | | |
| | Контакт-центр | + | - | - | + | | | | |
| | Вартість послуги (ключ) | 10$ | 5$ | 20$ | 10$ | | | | |
| | | | | | | | | | |

## 5.2 Технологічний аудит ідеї проекту

У таблиці 5.3 оціненомо можливість технологічної реалізації ідеї стартапу та показано технології, які можна застосувати для реалізації проекту.

Таблиця 5.3 –Технологічна здійсненність ідеї проекту

| № п/п | Ідея проекту | Технології реалізації | Наявність технологій | Доступність технологій |
|---|---|---|---|---|
| | Онлайн-кіно | Anroid | наявна | доступна |
| | Доступ через мережу | SIM карта | наявна | доступна |
| | Зв'язок клієнту с сайтом | SSL | наявна | доступна |
| | Персональний онлайн сервіс | Програмне забезпечення для ОС: Windows, Android, Mac | необхідно розробити | доступна |
| | Автономне живлення | ні | необхідно розробити | доступна |

## 5.3 Аналіз ринкових можливостей запуску стартап-проекту

У таблиці 5.4 показано попередню характеристику потенційного ринку стартап-проекту

Таблиця 5.4 -Попередня характеристика потенційного ринку

| № п/п | Показники стану ринку (найменування) | Характеристика |
|---|---|---|
| 1 | Кількість головних гравців, од | 8 |
| 2 | Динаміка ринку (якісна оцінка) | Зростає |
| 3 | Наявність обмежень для входу (вказати характер обмежень) | відсутні |
| 4 | Специфічні вимоги до стандартизації та сертифікації | NIST |
| 5 | Середня норма рентабельності в галузі (або по ринку), % | 34% |

У таблиці 5.5 показано характеристику потенційних клієнтів стартап-проекту

Таблиця 5.5 - Характеристика потенційнихклієнтівстартап-проекту

| Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|---|---|---|---|
| Необхідність шифрування інформації | Компанії що мають в своєму продукті шифрування трафіку. | Залежно від цільової групи продукт комплектується різного роду додатками для зручності користування. | - надійність<br>ресурсоефективність<br>- доступність<br>- простота |

| | | | |
|---|---|---|---|
| | | | - швидкість |

У табл. 5.6 наведено основні загрози реалізації стартап-проекту.

Таблиця 5.6 - Фактори загроз

| | Фактор | Зміст загрози | Можлива реакція компанії |
|---|---|---|---|
| | Незацікавленість клієнтів | Внаслідок невдалого маркетингу клієнт може не зацікавитись продуктом. | Внесення додаткових змін в реалізацію тазниження цін |
| | Втрата монополії | Втрата рангу єдиного гаранту якості технології | Якісне та кількісне нарощування інтенсивності |

У табл.5.7 наведено основні можливості під час реалізації стартап-проекту.

Таблиця 5.7 - Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---|---|---|
| 1.Монополія | Інноваційний тип послуг | Стандартизація на високому рівні |
| 2.Локальний | Відсутність єдиного постачальника послуг | Окремий підхід до кожної локальної ділянки |
| 3.Міжгалузева | Конкуренція з іншими галузями (постачальниками програмної і апаратної частини) | Необхідність співробітництва в окремих сегментах |
| 4.Товарно-видова | Подолання розсинхронізації відбувається за схожими технологіями, що реалізовані апаратно | За необхідності, використання приладів та програм схожого типу |
| 5.Цінова | Можливість заощадити за допомогою діагностики | Гнучка політика цін на доступ |
| 6.Марочна | Кожна діагностика має бути стандартизованою | Отримання монополії над стандартом синхронізації |

У таблиці 5.8 наведено особливості та вплив конкурентного середовища на впровадження проекту .

Таблиця 5.8 - Аналіз конкуренції в галузі за М. Портером

| Складові аналізу | Прямі конкуренти в галузі | Потенційні конкуренти | Постачальники | Клієнти | Товари-замінники |
|---|---|---|---|---|---|
| | Технологічні постачальники | Необхідність пошуку постачальників | Залучення малопопулярних постачальників | Незалежність у прийнятті клієнтських рішень | Надання переваги більш авторитетним технологічним рішенням |
| Висновки: | Незначна | Можливість виходу на ринок є | Постачальники диктують цінову політику на обладнання | Клієнти диктують вимоги до якості | Обмеження існують лише у разі відмови від діагностики |

У таблиці 5.9 проаналізовано конкуренцію проекта в галузі за М. Портером

Таблиця 5.9 - Обґрунтування факторів конкурентноспроможності

| | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|---|---|---|
| | Раціональніший ціновий показник | Можливість більш раціонально використати ресурсів |
| | Надання персональних інтеграційних послуг | Сервісна підтримка апаратної та програмної частини |
| | Синхронізованість | Синхронізація з усіма ОС. |
| | Спектр застосувань | Використання для ряду потреб користувачів. |

У табл. 5.10 наведено та обґрунтувано фактори конкурентноспроможності.

Таблиця 4.10 - Порівняльний аналіз сильних та слабких сторін Web-cinema

| | Фактор конкуренто-спроможності | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 3 | 2 | 1 | | | | |
| | Раціональніший ціновий коказник | 13 | | | | | | | |
| | Надання персональних сервісних послуг 24/7 | 15 | | | | | | | |
| | Синхронізованість | 20 | | | | | | | |
| | Спектр застосувань | 17 | | | | | | | |

У табл. 5.11 перелічено сильні та слабкі сторони проекту.

Таблиця 5.11 - SWOT- аналіз стартап-проекту

| | |
|---|---|
| Сильні сторони:, надання інтеграційних сервісних послуг, синхронізованість | Слабкі сторони: раціональніший ціновий показник |
| Можливості: використання для ряду потреб користувачів | Загрози: незацікавленість клієнтів, втрата монополії |

## 5.4 Розроблення ринкової стратегії проекту

Обгрунтування вибору цільових груп потенціійних споживачів показано в табл. 5.12 .

Таблиця 5.12 - Вибір цільових груп потенційних споживачів

| Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільовоїгрупи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|---|---|---|---|---|
| Розробники мікросхем | Середня | Високий | Високий | Середня |
| Розробники систем шфирування | Середня | Високий | Середній | Середня |
| Розробники ПЗ загального спрямування | Середня | Висока | Низький | Середня |

Визначення базової стратегії розвитку наведено у табл. 5.13.

Таблиця 5.13 - Визначення базової стратегії розвитку

| Обрана альтернатива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентос-проможні позиції від-повідно до обраної альтернативи | Базова стратегія розвитку |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Створення гаранту якості державного рівня | Встановлення єдиного універсального стандарту | Розробка і випуск власних пристроїв | Стратегія диференціації |
| Дешевизна проекту | Раціональніші витрати на обладнання, та послуги | Малові домі партнери з постачання обладнання | Стратегія лідерства по витратах |

Визначення основної стратегії конкурентної поведінки показано в табл. 5.14.

Таблиця 5.14 - Визначення базової стратегії конкурентної поведінки

| Чи є проект «пер-шопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конку-рентної поведінки |
|---|---|---|---|
| так | Забирати існуючих та шукати нових | Характеристики якості програмних реалізацій | Стратегія виклику лідера |

Визначення стратегії позиціонування показанов табл. 5.15.

Таблиця 5.15 - Визначення стратегії позиціонування

| Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкуренто-спроможні позиції власного стартап-проекту | Вибір асоціацій, які мають сфо-рмувати комплексну позицію власного проекту (три ключових) |
|---|---|---|---|
| Висока якість послуг | Стратегія диференціації | Синхронізова-ність | Якість, надійність, сервісність |
| Мінімальні витрати | Стратегія лідерства по витратах | Широкий спектр застосування | Дешевизна, раціональність, тех. підтримка |

## 5.5 Розроблення маркетингової програми стартап-проекту

Таблиця 5.16 - Визначення ключових переваг концепції потенційного товару

| Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючіаботакі, щопотрібностворити |
|---|---|---|
| Якість | Висока якість, сервісність | сервісність |
| Дешевизна | Раціональне використання коштів | дешевизна |

Виявлено три рівні моделі товару. Зміст та складові рівнів товару показано в табл. 5.17.

Таблиця 5.17 - Опис трьох рівнів моделі товару

| Рівні товару | Сутність та складові | | |
|---|---|---|---|
| I. Товар за задумом | Дешевий якісний товар та послуги, стандартизована якість послуг та обладнання | | |
| II. Товар у реальному виконанні | Властивості/характеристики: | М/Нм | Вр/Тх /Тл/Е/Ор |
| | 1) Варстість обслуговування,<br>2) Кількість елементів<br>3) Строк безвідмовної праці<br>4)Технологічна собівартість товару | 1) М<br>2) М<br>3) М<br>4) М | 1)Е<br>2) Пр<br>3)Нд<br>4)Тх |
| | Якість: дерстандарт якості, високоякісні технології | | |
| III. Товар із підкрі-пленням | До продажу – діагностика, обладнання, кріплення, дод.елементи живлення<br>Після продажу – персональний онлайн сервіс | | |

Завдяки чому потенційний товар буде захищено від копіювання: специфічна методика обробки даних.

Визначення цінової політики на послугу показано в табл. 5.18.

Таблиця 5.18 - Визначення меж встановлення ціни

| Рівень цін на послуги замінники | Рівень цін на послуги аналоги | Рівень доходів цільової групи споживачів | Верхня та нижняя межі встановлення ціни на товар/послугу |
|---|---|---|---|
| 50-200 у.о./п | 5-50 у.о./од | Середній | Н.10у.о. – В.30у.о. |

Створення системи збуту послуги вказано у табл. 5.19.

Таблиця 5.19 - Концепція маркетингових комунікацій

| Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламногоповідомлення | Концепція рекламного звернення |
|---|---|---|---|---|
| Зацікавленість в якісному продукті з раціональним використанням ресурсів | Мережні ресурси | Синхронізованість з будь-якими ОС | Зацікавити у покращенняхпов’язаних із зростаючою популярністю товару та послуг | Представлення продукції відправною точкою на шляху до безпеки |
| Зацікавленість у | Мережні ресурси | Широкий спектр | Зацікавити у позитивних | Представлення якісної |

| | | | | |
|---|---|---|---|---|
| великій кількості продукту із дотриман-ням умов якісності | | застосува-ння | сторонах | роботи зклієнтами |

**Висновки**

В цьому розділі проведено аналіз перспектив комерціалізації запропонованого стартап проекту щодо застосування та реалізації запропонованого рішення та виявлено, що на ринку послуг повязних з шифруванням та розробкою алгоритмів шифрування є достатній попит на дану пропозицію, яка якраз може його задовольнити.

Перспективність впровадження досить висока, адже основними групами клієнтів є крупні компанії що займаються розробкою електроніки та ПЗ, ів разі досягенення відповідного авторитету, існує можливість охоплення у масштабах міжнаціональних ринків. Конкурентноспроможність проекту забезпечує нижча ціна кінцевого продукту, менша необхідна кількість ресурсів та забезпечення високої якості шифрування в тих умовах, де конкуренти відстають за цим параметром. Це є перевагою і основним критерієм входження на ринок запропонованого рішення.

# ВИСНОВКИ

За результатами проведеного аналізу, викладеного в дисертації, були зроблені такі висновки:

1. Якісні псевдовипадкові послідовності, будучи по суті детермінованими, мають всі властивості реалізацій істинно випадкових процесів та успішно їх заміняють, оскільки випадкові послідовності надзвичайно важко формувати. Від якості використаних генераторів залежить якість отриманих результатів.

2. До генераторів псевдовипадкових послідовностей, та до самих згенерованих послідовностей існує ряд вимог та певних критеріїв якості. Для об’єктивної оцінки таких генераторів і послідовностей на відповідність цим вимогам було створено ряд тестів, найбільш розповсюдженими серед яких стали тести NIST.

3. Створенню хороших генераторів псевдовипадкових послідовностей приділяється достатньо велика увага в математиці. Всі генератори псевдовипадкових послідовностей при певних умовах дають передбачувані результати та кореляційні залежності. А це дозволяє крипто аналітикам здійснювати ефективні атаки на криптосистеми, де ці послідовності застосовуються. Як і будь-який криптографічний алгоритм, генератор криптографічно надійної псевдовипадкової послідовності може бути атакований і зламаний криптоаналітиком. В роботі розглянуто варіанти таких атак, а також методи, що дозволяють зробити такі генератори більш стійкими до криптоаналітичних атак різних типів.

4. В результаті проведеного дослідження був запропонований спосіб підбору полінома для ГПВП на основі регістру з лінійним зворотнім зв’язком по модулю два, а також представлено програмну реалізацію такого ГПВП.

5. Напрямом для подальших досліджень в даній області є розробка нових алгоритмів генерації ПВП, які дозволять ефективно протистояти атакам на потокові шифри, при цьому не потребують занадто великих ресурсів для

програмної чи апаратної реалізації. З цього випливає, що даний напрям досліджень є перспективним і потребує подальшого вивчення.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ


1. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. – М.: Мир, 1986. – 576 с., ил.
2. Ярмолик В.Н. Контроль и діагностика цифрових узлов ЭВМ. – Мн.: Наука и техника, 1988. – 240 с.
3. Huffman D.A., The Syntesis of Linear Sequential Coding Networks, Department of Electr. Eng. and Research Lab. of Electronics, M.I.T., Cambridge, Massachusetts (1955)
4. Берлекэмп Э. Алгебраическая теория кодирования. – М.: Мир, 1971, с.477
5. Удалов А.П., Супрун Б.А. Избыточное кодирование при передаче информации двоичными кодами. – М.: "Связь", 1964. – 269 с.
6. Яковлев В.В., Федоров Р.Ф. Стохастические вычислительные машины.– Л.: Машиностроение, 1974.–344 с.
7. С. М. Сухман, А. В. Бернов, Б. В. Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. — М.: Эко-Трендз, 2003.
8. С.М. Сухман, А.В. Бернов, Б.В. Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. – М.: Эко-Трендз, 2003.
9. Теория и применение псевдослучайных сигналов / А.И. Алексеев, А.Г. Шереметьев, Г.И. Тузов, Б.И. Глазов – М.: Наука, 1969. –367 с.
10. П. Хоровиц, У. Хилл. Искусство схемотехніки. В трех томах. – М.: Мир, 1993. – 2 т.
11. Б.В. Шевкопляс. Микропроцессорные структуры. Инженерные решения. Дополнение перове: Справочник. – М.: Радио и свіязь, 1993.
12. Исагулиев К.П. Справочник по криптологии. Изд. Новое знание, 2004 г, 237 стр.
13. Б. В. Шевкопляс. Микропроцессорные структуры. Инженерные решения. Дополнение первое: Справочник. — М.: Радио и связь, 1993

14. Европейский патент ЕР 0.340.694.А2.
15. Патент США № 5.530.959.
16. Патент США № 6.215.835 В1
17. Гумен Т.Ф., Малогулко Р.В., Савченко Ю.Г. Вплив інформаційної надлишковості на завадо захищеність та швидкість передачі повідомлень. "Електроніка і зв'язок" №6, 2007. – с. 85-88
18. Arvillias A.C., Maritsas D.G. Toggle-Resisters Generating in Parallel K kth Decimations of M-Sequences Design Tables.– IEEE Trans. on Computers, 1979, vol. C-28. N 2, p. 89-101
19. Arvillias A.C., Maritsas D.G. Combinational Logic Free Realisations for High–Speed Sequence Generation. – Electronics Letters, 1977, vol. 13, N 17, p. 500–502
20. Hurd W.J. Efficient Generation of Statistically Good Pseudonoise by Linearly Interconnected Shift Registers.– IEEE Trans. on Computers, 1974, vol. C-23, N 2, p. 146–152
21. Савченко Ю.Г., Малогулко Р.В. Вдосконалення генераторів ПВП та їх застосування в системах скремблер-дескремблер телекомунікаційних пристроїв. Наукові записки УНДІЗ, №6(8), 2008
22. Шеннон К., Теория связи в секретных системах, в кн."Работы по теории информации и кибернетике", Из-во ИЛ, М., 1963, с.333-402
23. Фількін К.М. Розробка методів порівняльного аналізу та оцінки алгоритмів формування сигналів для радіосистем з CDMA
24. Прикладная криптография [електронний ресурс] / Брюс Шнайер http://www.ssl.stu.neva.ru/psw/crypto/appl_rus/appl_cryp.htm

Додаток А

# ABSTRACT

на атестаційну магістерську дисертацію

" "

# ABSTRACT

In computing, a linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state.

The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value.

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appears random and has a very long cycle.

Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common.

The mathematics of a cyclic redundancy check, used to provide a quick check against transmission errors, are closely related to those of an LFSR.[

A pseudorandom sequencesgenerators application is extraordinarily wide. It is possible to distinguish, for example, next areas of their application:

- correction and exposing errors codes;
- digital devices and integrated circuitstesting;
- information protectio

High-quality pseudorandom sequences, being essentially determined, have all characteristics of truly random processes realizationand successfully them it is substituted for, as casual sequences it is extraordinarily difficult to form. Quality of the got results depends on quality of the used generators. This circumstance is

underlined by the known aphorism of Robert Cavu : "Generation of random numbers is too important, to abandon it on will of case". Finding the the real random numbers sourcesis difficult. Physical noises, such as detectors of changes of ionizing radiation, fractional noise in a resistor or space radiation, can be such sources. Such devices are however used in additions of network safety rarely. Rough attacks are also caused complication on similar devices. An alternative decision is some set from plenty of random numberscreation and publication of him in some dictionary. However, and such sets provide the very limited source of numbers in comparing to the that amount which needs to additions of network safety. Though sets from these tables really provide a statistical chance, they are casual not enough, as a malefactor can get the copy of dictionary.

Cryptographic additions use the special algorithms for genering of random numbers. These algorithms in are good time certain and generate the sequence of numbers, which in theory can not be statistically casual. At the same time, if a good algorithm is elected, then a numerical sequence will pass most tests on a chance. Such numbers are pseudocasual numbers.

Not a single determined algorithm can generate random numbers fully, he can only approximate some properties of random numbers. As John fon Neiman noticed, "every, who has a weakness to the arithmetic methods of random numbersreceipt, sinful out of every doubts".

Any pseudorandom generators with the limited resources "goes in cycles" sooner or later - begins to repeat the the same sequence of numbers. Length of cycles of pseudorandom generators depends on a generator and on the average folds approximately $2^{n/2}$, where $n$ is a size of the internal state in bats, though linear congruent and linear pseudorandom generators have a maximal cycle of order $2^n$. If pseudorandom generators is formed sequence goes to the very short cycles, then such pseudorandom generator becomes predictable and not suitable for practical application.

Most simple practical generators have high-rate though, but have many defects:

- too short period;
- successive values are not independent;
- some of bits "less casual", than other;
- uneven homogeneous distribution;
- reverseness.

For example, RANDU algorithm, that by decades used on mainframes (high-performance computer with the considerable main and external memoryvolume , intended for the centralized depositories of data of large capacity and implementation of intensive calculable works), appeared very bad, that caused doubts in authenticity of results of many researches which used this algorithm.

The linear congruent method most widespreaded, Fibonachi with a delay, shift register with a linear feed-back.

From modern pseudorandom generators wide application was also got by the "whirlwind of Mersen", offered in 1997 year byMatsumoto and Nisimura. Its advantages is an extraordinarily large period ($2^{19937}-1$), even distribution in a 623 measuring (a linear congruent method gives more or less even distribution at most in a 5 measuring), rapid random numbersgeneration (in 2-3 times quicker than standard pseudorandom generator, that use a linear congruent method). However, there are algorithms, which distinguish a sequence, formed by the "whirlwind of Mersen", as non-random. It does the "whirlwind of Mersen" not suitable for cryptography.

With an existent necessity to generate the easily repeated random sequences, also there is a necessity to generate unforeseeable or absolutely casual numbers fully. Such generators are the random generators named. As such generators are more frequent used for the generation of the unique symmetric and asymmetric keys for enciphering, they are mostly built from combination of cryptographically secure random generator andentropy outsourcing (just the same combination it is now accepted to understand under "random generator).

Almost all large producers of microchips supply with vehicle ГВЧ with different information generators, using different methods for their cleaning from an inevitable predictability. However now speed of random numbers collectiondoes not answer the fast-acting of modern processors all existent microchips (a few thousand bits in a second).

In modern computers the programmaticauthors of random generators use considerably more rapid "entropysources", such, as noise of sound card or step counter of processor. Informational entropy is a measure ofinformation chaoticness, vagueness of appearance of any symbol of primary alphabet. In default of informative losses numeral equals information content on the symbol of report which is passed. Before appearance of possibility of read-out of values of meter of times, collection of information from the "sources of entropy" was the most vulnerable mestome of random generator. This problem and until now fully not decided in many devices (for example, smart-cards) which remain vulnerable thus. Much random generators use traditionally tested, though slow, methods of collection of information from the "sources of entropy" like measuring of reaction of user (motion of manipulator of type is a "mouse" and other), as, for example, in PGP (computer program, that allows to execute the operations of enciphering (code) and digital signature of reports, files and other information, presented in a digital kind) and Yarrow (numerical generator which presently is not used), or co-operations between threads, as, for example, in Java of secure random.

The variety of pseudorandom characters generator is pseudorandom bits generator (PRBG) are generators of pseudorandom bits, and also thread codes. pseudorandom bits generator , as well as current codes, consist of the internal state (usually, by a size from 16 bits to a few the megabyte), functions of initialising internal will become the key (seed), functions of updating of the internal state and function of leadingout. Pseudorandom bits generator  divide into simple arithmetic, broken cryptographic and cryptographically secure. Them a common purpose is a generation of sequences of numbers, which it is impossible to distinguish from casual calculable methods.

Although much pseudorandom characters generator or thread codes offer "much random" numbers far, such generators are considerably more slow ordinary arithmetic and can be useless in different researches which require, that a processor was at leisure for more useful calculations.

In soldiery aims and in the field terms used synchronous cryptographically secure of pseudorandom characters generator (thread codes) is only secret, sectional codes are not used. The examples of known cryptographically secure of pseudorandom characters generators are RC4, ISAAC, SEAL, Snow, quite slow theoretical algorithm of Bloom, Bloom and fur Coat, and also meters with cryptographic hesh-fuctions or crypto systained by sectional codes instead of function of leadingout.

Some examples of the encipheringcryptographically secure:

Cyclic enciphering. In this case the method of generation of the key of session is used from the master key. A meter with the period of N is used as included in an enciphering device. For example, in case of the use of the 56-bit key of DES used meter with a period 256. After each створенного key of value of meter increases on Thus, a pseudocasual sequence, got on this chart, has a complete period: every initial value of $X_0$, $X_1$,…$X_{N-1}$ it is based on the different values of meter and, to the volume, $X_0 \neq X_1 \neq X_{N-1}$. So as a master-key is secret, it easily to show that no secret key depends on one knowledge or more previous secret keys.

ANSI X9.17. Pseudorandom generator from the standard of ANSI X9.17 used in many additions of financial safety and PGP. In basis this PRNG lies triple DES. Generator of ANSI X9.17 consists of next parts:

Entrance: a generator is managed by two pseudocasual entrances. One is to 64-bit presentations of current date and time, which change each time at creation of number. Other is 64-bit by an initial value. It will be initialized by some arbitrary value and changes during generuting of sequence of pseudocasual numbers.

Keys: a generator uses three modules of triple DES. All three use the same pair of the 56-bit keys, which sticks to in a secret and used only at generuting of pseudocasual number.

Return: a return consists of 64-bit pseudocasual number and 64-bit value which will be used as an initial value at creation of next number.

- $DT_i$ is a value of date and time on beginning of i of the th stage of generating.
- $V_i$is an initial value for i of the th stage of generuting.
- $R_i$is a pseudocasual number, created on i to the th stage of generuting.
- $K_1$, $K_2$are the keys, used on every stage.

A chart includes the use of the 112-bit key and three EDE encryption. On an entrance two pseudocasual values are given: value of date and time and initial value of current iteration, on a return get an initial value for a next iteration and duty pseudocasual value. Even if the pseudocasual number of $R_i$ will be compromised, to calculate $V_{i+1}$from $R_i$ is not possible for clever time, and, thus, next pseudocasual value of $R_{i+1}$, so as for the receipt of $V_{i+1}$three operations of EDE are additionally executed.

Except out-of-date, well known LFSR generators, widely used as vehicle pseudorandom generators in XX century, unfortunately, it is very small known about modern vehicle pseudorandom generators (current codes), as majority from them is worked out for soldiery aims and sticks to in a secret. Almost all existent commercial vehicle PRNG is patented and also stick to in a secret. Vehicle PRNG is limited to the severe requirements to expense memory, id est "annex memory" which is used by a generator for creation of sequence (more frequent than all use of memory it is forbidden in general), by a fast-acting (1-2 times) and by an area (several hundred FPGA - or ASIC of -комірок). From such severe requirements to vehicle PRSG it is very difficult to create a криптостійкий generator, that is why to this day all are known vehicle PRSG were broken. The examples of such generators are Toyocrypt

and LILI - 128, that is LFSR generators, and both were broken by means of math attacks.

About good vehicle pseudorandom generators lack producers are forced to apply considerably more slow, but sectional codes (DES, AES) and hash-functions (SHA - 1) are well-known in the thread modes.

Pseudocasual sequences also fold basis of technology of CDMA (Code Division Multiple Access) - technologies many station accesses with the code demultiplexing. They provide expansion of spectrum code demultiplexing. Expansion of spectrum is conducted due to modulation of bearing oscillation by law of pseudocasual sequence, the direct method of modulation (direct sequence) and modulation the saltatory switching of frequency (frequency hopping) are here used. A code division or distinction of channels in the system from CDMA is carried out due to an appropriation to every subscriber channel such code PRC, which by maximal character uncorrelated with the signature sequences of other subscriber channels. In majority of CDMA the system of synchronization between the base and subscriber stations is also provided by means of pseudocasual sequences. It can be both signature, pilot-signal sequences are and specially distinguished with the small values of lateral extrass them autocorrelation functions.

Among the known families of random sequences of length $2^N - 1$ with near to ideal autocorrelation of most distribution in wideband connection purchased m-sequences, as a generation of these sequences is most simple, and their properties as compared to other are studied far better. Presently in not alone hundred works are the world counted for m-sequences, but the personal interest does not weaken to them. However, being linear, m-sequences is characterized by the small value of linear complication. This defect some other sequences are deprived as Hadamard and, foremost, sequence of GMW, curiosity to which considerably grew in the last time. The quantity of family of sequences of GMW at the large values of $N$ in oftentimes exceeds the number of m-sequences. The construction of such random sequences substantially extends an initial base for forming of maximal on a volume subsets of

random sequences with the acceptable level of cross-correlation, that allows in one cases to increase the number of users at set noise immunity, and in other - to reduce a mutual noise level at the fixed number of users.

The first systems of generators of sequences of GMW were built and described as early as 70th. Their principle of work was based on decouplig property of sequences of GMW. On this account all these generators got the name of decouplig. A basic difference between them consists in an amount forms which are generated. It is necessary to mark, that all these generators are characterized by the exponensial increase of complication of realization depending on $N$ and that is why large practical distribution they did not get. At the same time, the indisputable utility of decouplig generators appeared in that they served a prototype at creation of one of the most simple generators. Yes, at 1984 Welsh and Sholls there was the offered method of generation of classes of sequences of GMW, that is based on the generation of $q$ th m-sequence. This publication became beginning of triumphal distribution of sequences of GMW and at the same time gave a mighty impulse for their further research. In future it was shown by the same authors, that this method can be widespread on all classes of such sequences. Unfortunately, on different reasons, pioneer works remained unknown.

Further decades introduced nothing cardinal in the technique of generation of sequences. A substantial breach took place only in 1997 year, when at Scientific and technical conference the new method of generation of binary sequences of GMW was offered, which simplified development of generators substantial character. Simplicity of this method unlike the method of Велча-Шольца consists in the use of the moved copies of binary m-sequence and, as a result, foresees the use only of binary logic. Therefore there are grounds to expect that with appearance of such generators number of developers of communication networks which use the sequences of GMW will increase.

Digital logical devices it easily to organize as circles of shift registers, imitating circular shifts and polinom arithmetic, that used for description of cyclic

kodes. Therefore the structure of cyclic kodes is closely related to the structure of circles of shift registers. These circles to my surprise successfully walked up for realization of many procedures of code and decoding, in which they assume an air of filters often. Many algorithms it is simpler to describe, using the symbolics of circles of shift registers. Many details it more easily to understand on a chart, than on a formula.

Shift registers useful to development theories, as they act part pseudomathematical denotations which help better to understand some actions above polynomials.

A synthesis of generators of pseudocasual test sequences is important in the field of the probabilistic testing. In present time a signature analysis is most often used in the new technique of testing of digital charts. He serves as the method of exposure of errors in the sequence of data for an analysis, which are caused by the disrepairs of control digital device. By forming of test sequence on the entrances of digital device for an analysis for each it is possible to find the standard value of signatures, the great number of which is memorized and, in future, used for comparing to the value of signatures which are removed from devices for verification, its pole. Any deviation of the really got signature from standard testifies that the pole of chart functions differently from the case of the in good condition state of device. This procedure almost repeats procedure of being of disrepairs in analog devices, that consists in the successive measuring and analysis of some analog quantities.

Basic advantage of signature analysis is simplicity of its application for determination and localization of disrepairs of digital charts, as the sophisticated stand apparatus absents during realization of test experiment and minimum skills are needed only for its realization.

Random numbers sensors which are built into compilers not suitable for cryptographic additions, as numbers are generated by them casual not enough. Large enough attention is spared creation of good generators of pseudocasual sequences in

mathematics. For today it is succeeded to create sequences with the period of order 2-3 Kbits.

Another useful aspect offered approach is related to possibility of simultaneous priv, which is passed (kept) from an unauthorized division. Level of security here not very much high as compared to the use of modern methods of enciphering, but at least near to defence by means of passwords. This possibility can present practical interest in some partial cases, when time during which the passed report remains actual is small, and surplus of variants can be realized quickly enough, for example, hardwarily. It is possible also to mark noise immunity of transmission. However, obviously, her level will fully depend only on surplus of initial great number of reports.

In cryptography to the pseudocasual sequences greater requirements, than simple presence for them of certain terms of statistical chance, are pulled out. That a pseudocasual sequence was cryptographic reliable, it is necessary, that it was unforeseeable. It means that for cryptographic reliable pseudocasual bit sequence it is impossible in good time to say, which will be it next bit, even knowing the algorithm of generation of this sequence and all it previous bats. As well as any cryptographic algorithm, generator cryptographic reliable pseudocasual sequence can be attacked and broken by a крипто analyst.

There are methods on creation of such generators proof to chrypto analytical attacks of different types.

In general case there is not a simple method of generuting of primitive polynomials certain to the degree. Simpler all to elect a polynomial casual character and to check, whether he is not primitive. Most packages of mathematical softwares are able to decide such task.

Increase of period of generuting of pseudocasual numbers more than to $2^n$ is not possible. But due to introduction of additional lanocs to such generators it is possible to increase the period of generuting, which will depend only on the amount of the found primitive polynomials of identical bit.

For the sequence of linear digital фільтра with reverse copulas on the module two lengths $n$ the internal state is previous $n$ initial bits of generator. Even if the chart of feed-back is kept in a secret, it can be certain for $2n$ to the initial bats of generator by means of some high-efficiency algorithms.

Presented results, got at research of influence of informative surplus on passed to information and her property, and also a few models are presented for a compression and proceeding in information for shift registers with reverse copulas and a few improved models of generators of pseudorandom sequences on these registers.

Other boolean functions two variables are required to the analysis of got pseudorandom sequence "on a chance". Essentially, a language will go about verification of appearance in pseudorandom sequence of the easily forecast conformities to law.